Nou esquema segur de firma basat en atributs

Escola Tècnica Superior de Telecomunicacions de Barcelona

Firmat,

$$\sigma = \Big(\{C_i, \pi_i, Com(f_i), Com(\tilde{\lambda}_i), Com(\lambda_i), \pi_{f_i}, \pi_{\lambda_i}, \pi_{\mu_i}\}_{\mathsf{at}_i \in B}, \{\pi_{\psi_k}\}_{k=1}^{h}, \sigma_1, \sigma_2, Com(\sigma_3), Com(K), Com(\sigma_K), \pi_K, \pi_\sigma \Big)$$

Alex Escala Ribas
Directora: Dra. Paz Morillo Bosch

Departament de Matemàtica Aplicada IV
Juliol 2010

Agraïments

Primer de tot, voldria agrair a la Paz tot el suport que m'ha estat donant des del primer dia i tota la confiança dipositada en mi. També li voldria agrair les hores dedicades en aquest projecte, que no són poques, i tots els moments en els quals m'ha animat quan estava encallat en algun tema. A banda de tot el que fa referència al projecte, també m'ha ensenyat i ajudat moltes coses i el mínim que puc fer és donar-li les gràcies.

També voldria donar les gràcies als meus companys del departament per un ambient de treball immillorable. En particular, voldria donar les gràcies a dues persones que m'han ajudat molt en aquest projecte. La primera és el Javi, qui em va donar moltes idees que em van portar a obtenir l'esquema de firma i també va fer una correcció molt curosa del capítol en el qual es presenta el resultat. La segona és la Carla, que em va posar sobre la pista de les proves Groth-Sahai i m'ha anat aportant moltes reflexions i idees realment útils.

No hauria estat possible arribar fins on sóc sense l'ajut de moltes persones. Per això, voldria donar gràcies als meus companys i amics de la universitat, amb qui he compartit molts bons moments tant de festa com d'estudi. En concret, voldria agrair a l'Anna i al Carlos els 5 anys i mig que hem compartit; sense ells tot hauria estat ben diferent.

Tampoc hauria arribat fins aquí sense el CFIS, que m'ha donat l'oportunitat de fer una doble titulació que m'ha aportat moltes coses que no hauria estat capaç d'obtenir de cap altra manera. Voldria donar les gràcies a totes les persones del CFIS que en algun moment o altre m'han anat ajudant en el meu pas per la universitat.

Finalment, però no amb menys importància, voldria agrair a la meva família i en especial a l'Eli tot el suport que m'han donat. Haver-los d'explicar el meu projecte sense tecnicismes matemàtics m'han ajudat a aprendre com fer entendre a qualsevol persona del que tracta el meu projecte, cosa que considero molt important.

Índex

Introducció

Sobre les firmes digitals

Les firmes digitals són una eina criptogràfica bàsica amb moltes aplicacions i extensions. Els esquemes de firma digital sorgeixen a partir dels criptosistemes de clau pública, introduïts l'any 1976 per Whitfield Diffie i Martin Hellman en un article titulat *New directions in Cryptography* [DH76]. La idea d'una clau privada, que només coneix un usuari, i una clau pública, coneguda per tothom, va permetre idear esquemes en els quals la clau privada es fa servir per signar missatges i la clau pública serveix per verificar la validesa de la firma.

Una firma digital té dues propietats bàsiques. Per una banda, una firma garanteix autenticitat. Això vol dir que un missatge firmat permet identificar-ne el firmant i assegurar que només ell l'ha pogut firmar. L'altra propietat és la integritat: una firma assegura que el missatge no ha estat modificat després de que s'hagi realitzat la signatura. Tot i que aquestes propietats ja les tenim amb les firmes a mà, les firmes digitals ens donen una seguretat molt més gran: falsificar una firma manual pot arribar a ser relativament fàcil, però falsificar una firma digital és gairebé impossible si no es coneix la clau secreta del firmant.

La seguretat d'una firma digital es basa en problemes que són computacionalment difícils. Això vol dir que farien falta milions d'anys per trencar la seguretat d'una firma. Ara bé, la noció de seguretat també s'ha de definir a fons: s'ha de definir què s'entén per trencar un esquema de firma. Per això es defineixen diferents nivells de seguretat, que comentarem més endavant.

En l'actualitat, les firmes digitals s'usen bastant sovint. Quan ens connectem a Internet, sabem que ens estem connectant a una web de confiança gràcies a les firmes digitals. També tenim la possibilitat de firmar documents digitalment amb el DNI electrònic, doncs aquest conté la clau secreta i la clau pública per fer-ho.

Les diferents eines matemàtiques que existeixen permeten realitzar variants de les firmes digitals amb altres propietats interessants. Un exemple són els esquemes de firmes basats en identitats. En aquest tipus d'esquemes, la clau de verificació de les firmes és la identitat del firmant. En una firma manual o una firma digital bàsica, necessitem saber a

qui correspon la firma. En una firma basada en identitats, això és immediat. El firmant aconsegueix la seva clau secreta d'una autoritat, és a dir, una entitat de confiança que distribueix les claus secretes un cop ha identificat el firmant.

Les firmes digitals també permeten extensions impossibles de realitzar amb firmes manuals. Probablement les extensions més sorprenents de totes són les que tenen a veure amb l'anonimat: es poden construir esquemes de firma digital en els quals donada la firma no se sàpiga qui és el firmant però sí que se'n desprengui alguna informació sobre les característiques del firmant. Per exemple, en les firmes de grup un participant firma un missatge en nom d'un grup de participants. De la firma se'n desprendrà que alguna persona dins d'un grup ha firmat el missatge. Tot i que els membres del grup són coneguts, qualsevol d'ells pot haver realitzat la firma i ningú podrà esbrinar qui ha estat.

Aquests esquemes avançats de firma digital tenen una gran quantitat d'aplicacions, tot i que en l'actualitat no deixen de ser unes eines purament teòriques. Les dificultats de la indústria d'adaptar-se als avenços que es van produint, sumades a les dificultats de la societat de confiar en noves eines de seguretat fan que es trigui molt en posar a la pràctica els esquemes. Per exemple, l'any 1978 apareix el primer esquema de firma digital, però no és fins l'any 2006 que el DNI electrònic incorpora aquestes facilitats.

La nova proposta

En aquest treball proposem un nou esquema de firma basat en atributs. La idea bàsica d'un esquema de firma basat en atributs és que, enlloc de firmar com a una identitat es firma com a algú que conté una sèrie d'atributs. Un firmant demanarà claus secretes pel conjunt d'atributs que té i firmarà el missatge usant aquestes claus. Per exemple, jo podria firmar aquest treball com a una persona que és (1) estudiant i (2) menor de 25 anys. Quan algú volgués comprovar la firma, no podria deduir que sóc jo qui ha realitzat la firma sinó una persona amb aquestes dues propietats. De fet, una firma basada en atributs permet definir conjunts d'atributs que es poden usar per firmar. Dit d'una altra manera, una firma pot estar realitzada per algú que contingui un subconjunt d'atributs d'un conjunt donat. Seguint l'exemple d'abans, jo podria firmar un missatge com a algú que té tres dels quatre atributs següents:(1) ser un estudiant, (2) ser menor de 25 anys, (3) estar empadronat a les Illes Balears i (4) tenir llicència per conduir. Donada la firma, ningú podria saber quins tres atributs posseeixo. Això podria ser útil en el cas que volgués optar a una beca que demanessin tenir tres d'aquestes característiques i no interessés saber exactament les característiques de cada persona per evitar discriminacions.

Per aconseguir aquest esquema ens hem basat en dues coses: per una banda hem modificat un esquema de firma proposat per Shacham-Waters [SW07] i per altra banda hem utilitzat les proves de Groth-Sahai [GS07]. Pel que fa a l'esquema de firma en el qual ens basem, l'interès és simplement l'existència d'aquest i les demostracions de seguretat, que es poden adaptar al nostre esquema. D'altra banda, les proves de Groth-Sahai són

una eina molt poderosa al mateix temps que complexa, per tant hem decidit dedicar un capítol sencer d'aquest treball per explicar-les. L'estudi d'aquesta eina ha estat una part important del projecte de final de carrera.

Les proves de Groth-Sahai permeten fer proves sense coneixement de l'acompliment d'un conjunt d'equacions. El concepte de proves sense coneixement (traduït lliurement de l'anglès, Zero-Knowledge Proofs) és segurament un dels conceptes més complicats d'entendre del món de la criptografia. En aquest treball dediquem un capítol a explicar què és una prova sense coneixement, quina és la intuïció darrera d'aquestes proves i quines propietats han de complir.

Estructura del treball

El treball s'estructura de la següent manera: en el capítol 1 s'expliquen els coneixements previs necessaris per entendre la resta del treball. Concretament, s'expliciten els algorismes de firma digital i es defineix amb detall el concepte de seguretat en un esquema de firma. També es dóna una petita introducció al concepte de grup bilineal, que s'utilitza en tot el treball.

En el capítol 2 s'explica el que s'entén per una prova sense coneixement. Tot i que també es pot considerar un coneixement previ, és prou important com per dedicar-li un capítol. A més, hem considerat que en la literatura sobre el tema no hi havia cap treball en el qual aquest concepte s'expliqués començant per l'aspecte intuïtiu i arribant a la formalització completa.

En el capítol 3 s'explica l'eina que utilitzem en la construcció de l'esquema: les proves de Groth-Sahai. Es tracta d'un sistema de proves no interactives que permeten demostrar l'acompliment d'equacions algebraiques en grups bilineals. Es manté l'eficiència de sistemes anteriors però ampliant el camp d'aplicació.

Finalment, en el capítol 4 es presenta el resultat aconseguit en aquest treball: una proposta de firma basada en atributs amb millores substancials respecte als esquemes existents en la literatura. Aquest capítol és la nostra contribució original i es basa en tot el treball presentat en els capítols anteriors.

Capítol 1

Coneixements previs

1.1 Grups bilineals

En tot aquest treball usarem el que anomenem grups bilineals.

Un grup bilineal és un conjunt de tres grups $(\mathbb{G}_1, \cdot), (\mathbb{G}_2, \cdot), (\mathbb{G}_T, \cdot)$ i una aplicació bilineal e. El símbol $\cdot$ denota l'operació del grup. Tot i que d'ara endavant usarem notació multiplicativa en la majoria del treball, a vegades s'usa notació additiva per $\mathbb{G}_1$ i $\mathbb{G}_2$ L'ordre dels tres grups ha de ser el mateix i pot ser o bé primer o bé producte de dos primers diferents.

L'aplicació e es defineix com una aplicació $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ amb les següents propietats:

- **Bilinearitat**: per tot $g_1 \in \mathbb{G}_1, g_2 \in \mathbb{G}_2, a, b \in \mathbb{Z}$ s'ha de complir que

$$e(g_1^a, g_2^b) = e(g_1, g_2)^{ab}.$$

- **No degeneració**: si g_1 és un generador de $\mathbb{G}_1$ i g_2 és generador de $\mathbb{G}_2$, aleshores $e(g_1, g_2)$ és un generador de $\mathbb{G}_T$.

- La funció e ha de ser **eficientment computable**.

Un exemple de grup bilineal és prendre $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ com el grup format per punts de corbes el·líptiques i una modificació del pairing de Tate [Men05] o del pairing de Weil com a aplicació bilineal [Jou02]. En tot cas, existeixen grups bilineals tant amb ordre primer com amb ordre compost i tant amb $\mathbb{G}_1 = \mathbb{G}_2$ com $\mathbb{G}_1 \neq \mathbb{G}_2$.

Durant aquest treball farem servir la propietat bilineal del pairing, que en particular implica que

$$e(g_1^a, g_2) = e(g_1, g_2)^a = e(g_1, g_2^a)$$

1.2 Firma digital

Una firma digital és una eina criptogràfica que s'utilitza per demostrar l'origen i la integritat d'un missatge. En una firma digital hi actuen quatre participants o algorismes: el d'inici, el generador de claus, el firmant i el verificador.

- **Inici:** L'algorisme d'inici pren com a entrada un paràmetre de seguretat 1^k i obté a la sortida (mpk, msk), on mpk és informació pública d'inici com ara un grup bilineal i msk és informació secreta com ara la factorització de l'ordre del grup.
- **Generació de claus:** L'algorisme de generació de claus pren per entrada (mpk, msk) i obté a la sortida un parell de claus (s_k, v_k), que són la clau de firma i la clau de verificació.
- **Firma:** L'algorisme de firma pren com a entrada (mpk, s_k, v_k, msg), on msg és el missatge a firmar, i genera una firma σ.
- **Verificació:** L'algorisme de verificació pren com a entrada (mpk, σ, msg, v_k) i treu un 1 a la sortida si la firma és vàlida i un 0 altrament.

Depenent de l'esquema, podem suposar que un mateix participant duu a terme els protocols d'inici, generació de claus i firma. Ara bé, en esquemes on els tres protocols els duen a terme entitats diferents podem obtenir una major flexibilitat.

Tot i que inicialment la firma digital servia només per verificar que un missatge l'ha firmat una persona amb una clau de verificació concreta, han aparegut diversos esquemes que modifiquen la idea de firma digital. En el capítol 4 n'explicarem alguns en detall.

1.2.1 Seguretat d'una firma digital

Una de les característiques de l'esquema de firma proposat en aquest treball és que millora la seguretat dels esquemes existents. Per entendre què vol dir que un esquema és segur, o que un esquema és més segur que un altre, anem a explicar com demostrem la seguretat dels esquemes criptogràfics. Goldwasser, Micali i Rivest [GMR88] van ser els primers en definir formalment la noció de la seguretat d'una firma digital. A continuació intentarem explicar com defineixen la seguretat.

Joc de seguretat

Una demostració de seguretat ve caracteritzada per un adversari $\mathcal{A}$ que intenta trencar alguna part de la seguretat de l'esquema. Per exemple, l'adversari pot intentar falsificar una firma digital o, quan intervenen conceptes d'anonimat, l'adversari pot intentar

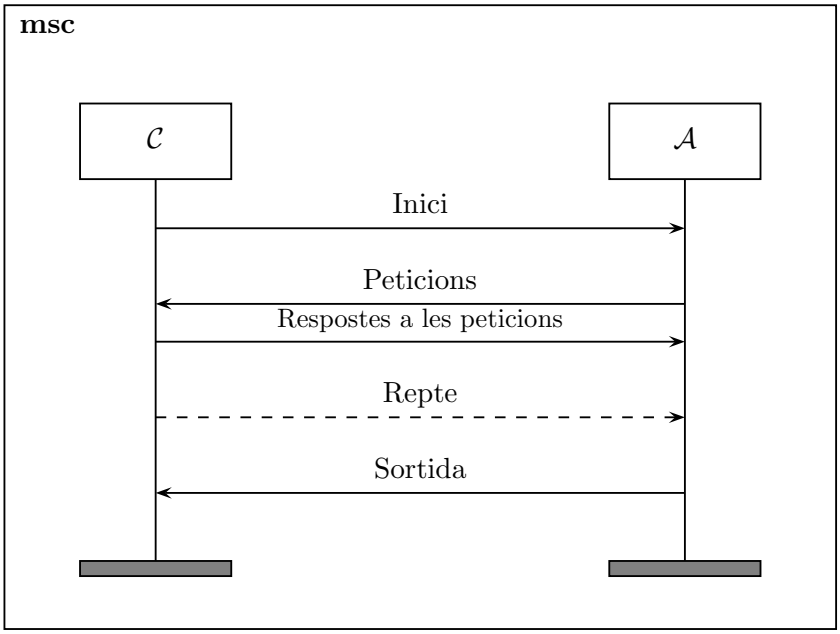


Figura 1.1: *Joc de seguretat entre l'adversari $\mathcal{A}$ i el reptador $\mathcal{C}$*

distingir qui ha realitzat la firma. En esquemes de firma, sempre suposarem que l'adversari té capacitat computacional limitada i en conseqüència parlarem sempre de seguretat computacional.

Per demostrar que un esquema té alguna propietat de seguretat, es fa el que anomenem un joc de seguretat. Aquest joc està dut a terme entre l'adversari i un reptador, qui tria tots els paràmetres de l'esquema, interactua amb l'adversari i el repta a trencar alguna part de la seguretat de l'esquema. Aquest joc consta, en general, de quatre passos: l'inici, les peticions, el repte i la sortida. La figura 1.1 il·lustra aquest joc.

En l'**inici**, el reptador escull els paràmetres del sistema. A vegades ho fa tenint en compte algunes dades que té a priori, però això ho comentarem més endavant. El reptador li dóna a l'adversari la informació que en l'esquema és pública. A més a més, a vegades li dóna certa informació privada. Això ens permetrà dir que l'algorisme té la propietat de seguretat que estem analitzant, encara que l'adversari obtingui aquesta informació secreta. En un escenari real, probablement serà difícil que algun adversari pugui obtenir aquesta informació secreta, però molts cops la incloem en la demostració perquè és senzill fer-ho i així aconseguim un nivell de protecció més gran.

En les **peticions**, l'adversari fa una sèrie de peticions al reptador. El tipus de peticions que pot fer ve definit pel joc de seguretat. Exemples de peticions que pot fer són peticions de claus secretes de firma o peticions de firmes de missatges. En esquemes de firma es distingeixen tres casos. En el cas que a l'adversari no se li permeti fer cap petició de firma ni de clau secreta es parla de seguretat sota un atac de coneixement de clau (en

anglès, Key-Only Attack). El segon cas és quan l'adversari pot fer peticions de firmes però sense decidir quins són els missatges (és a dir, els missatges a firmar els tria el reptador), aleshores parlem de seguretat sota un atac de missatge conegut (en anglès, Known Message Attack). Evidentment això implica que tampoc es poden fer peticions a clau privada de firma perquè donada la clau privada el reptador pot firmar qualsevol missatge que vulgui. Finalment, el tercer cas és quan l'adversari sí que pot triar els missatges dels quals obtindrà una firma per part del reptador. En aquest cas parlem de seguretat sota un atac de missatge escollit (en anglès, Chosen Message Attack). En aquest cas a vegades se sobreentén que les peticions de firmes poden ser adaptatives, és a dir, que cada petició es fa tenint en compte les anteriors, tot i que no sempre ha de ser així.

El **repte** és un pas que no sempre apareix en els jocs de seguretat. En aquest pas, el reptador li dona a l'adversari un repte. En una demostració de seguretat d'un esquema de firma amb anonimitat, el repte pot ser una firma i dos possibles firmants, i l'adversari ha de decidir qui ha realitzat la firma. Per altra banda, en la demostració de que una firma és no falsificable, aquest repte no sempre hi és ja que l'adversari el que ha de fer és falsificar una firma i el missatge a firmar no té perquè venir del reptador.

Finalment, l'adversari produeix una **sortida**. Aquesta sortida està relacionada amb l'objectiu de l'adversari. En l'exemple de l'esquema amb anonimat que hem explicat abans, la sortida pot ser el firmant que ha realitzat la signatura digital. En un esquema de firma, en el joc de no falsificació, la sortida normalment és una falsificació (o la clau privada directament). Quan parlem de falsificació, tornem a distingir quatre casos. El primer cas és quan la falsificació és d'un missatge escollit pel reptador. En aquest cas es parla de seguretat universal. Si el missatge és escollit per l'adversari abans de l'inici del joc, es parla de seguretat selectiva. Finalment, si l'adversari escull el missatge en el moment de fer la firma, es parla de seguretat existencial. Un cas especial és quan enlloc de produir una falsificació s'aconsegueix la clau secreta. Aleshores es parla d'una ruptura total. De la mateixa manera que considerem com és el missatge a la sortida també podem considerar altres paràmetres. Per exemple, si en un esquema de firma basada en identitats la identitat de la falsificació l'escull l'adversari abans del joc, es parla d'identitat selectiva.

Diem que l'adversari té èxit quan aconsegueix el seu propòsit. En el cas del joc de no falsificació, l'èxit s'aconsegueix al falsificar una firma. En el joc, molts dels algorismes que hi apareixeran seran probabilístics. L'inici de l'esquema pot ser-ho, com també ho seran les peticions dels missatges i/o el repte. Per tant, parlarem de la probabilitat d'èxit de l'adversari. Direm que un esquema té la propietat de seguretat analitzada quan la probabilitat d'èxit de l'adversari sigui negligible respecte algun paràmetre de seguretat.

Nivells de seguretat

En la secció anterior hem vist els diferents casos que podem considerar en la seguretat d'una firma digital. Ara bé, quin és el cas que proporciona millor seguretat?

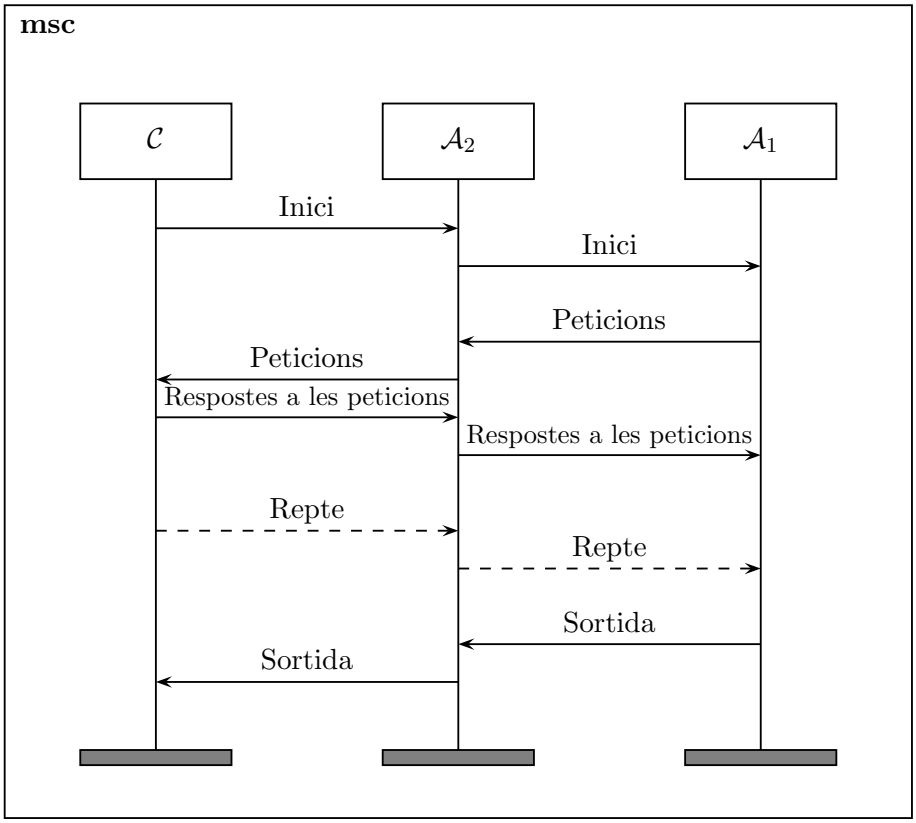


Figura 1.2: *Dos jocs de seguretat: entre $\mathcal{A}_2$ i $\mathcal{C}$ i entre $\mathcal{A}_1$ i $\mathcal{A}_2$*

La idea intuïtiva és que com més flexibilitat atorguem a l'adversari, més facilitats tindrà en el moment de fer la falsificació. Si ens protegim contra un adversari amb totes les facilitats, també estarem protegits contra adversaris amb menys facilitats. Pel que fa a les peticions, la major flexibilitat s'aconsegueix permetent a l'adversari demanar firmes per missatges de la seva elecció. Pel que fa a la sortida, el més fàcil per l'adversari serà si acceptem que la falsificació sigui sobre qualsevol missatge triat per l'adversari després de les peticions.

En criptografia, la relació entre els diverents nivells de seguretat es formalitza realitzant reduccions entre els casos. La figura 1.2 il·lustra la la redducció entre els casos. Per explicar-ho clarament usarem un exemple de seguretat existencial de no falsificació d'un esquema de firma.

Considerem dos escenaris: en el primer escenari volem seguretat sota un atac de missatge conegut i en el segon escenari volem seguretat sota un atac de missatge escollit. Tindrem un adversari $\mathcal{A}_1$ del primer escenari i un adversari $\mathcal{A}_2$ del segon escenari. Suposem que $\mathcal{A}_1$ és capaç de guanyar el joc de seguretat amb certa probabilitat. $\mathcal{A}_2$ guanyarà el joc quan $\mathcal{A}_1$ ho faci interactuant amb ell de la següent manera. Hem de tenir en compte que són dos jocs de seguretat diferents. En el joc de seguretat de $\mathcal{A}_2$, hi ha un reptador $\mathcal{C}$

extern i l'adversari $\mathcal{A}_2$. En el joc de seguretat de $\mathcal{A}_1$, el reptador serà $\mathcal{A}_2$. Quan $\mathcal{C}$ executi l'inici i li doni la informació a $\mathcal{A}_2$, $\mathcal{A}_2$ li passara la mateixa informació a $\mathcal{A}_1$. En el torn de peticions, $\mathcal{A}_2$ farà peticions de firma per una sèrie missatges (recordem que en aquest escenari li permetem fer-ho). $\mathcal{A}_2$ li donarà a $\mathcal{A}_1$ les parelles firma-missatge (recordem que en aquest cas estem sota un atac de missatge conegut). Finalment $\mathcal{A}_1$ aconseguirà una falsificació, que li passarà a $\mathcal{A}_2$, qui usarà la mateixa falsificació per guanyar el joc.

És clar que quan $\mathcal{A}_1$ faci una falsificació, $\mathcal{A}_2$ també la pot aconseguir. Per tant, la probabilitat que $\mathcal{A}_2$ tingui èxit és major o igual que la probabilitat d'èxit de $\mathcal{A}_1$. Per tant, si aconseguim que la probabilitat d'èxit de $\mathcal{A}_2$ sigui negligible respecte a algun paràmetre de seguretat, també ho serà la probabilitat d'èxit de $\mathcal{A}_1$.

Amb això hem aconseguit establir un ordre de nivells de seguretat. En una firma digital, el nivell de seguretat òptim és aconseguir un esquema amb no falsificació existencial sota atac de missatge escollit (en anglès, EUF-CMA, de Existential UnForgeability under Chosen Message Attack).

L'esquema que proposem en aquest article és existencialment no falsificable sota atac de missatge escollit. A diferència dels altres esquemes de firma basats en atributs, no demanem que la política d'atributs, és a dir, les possibles famílies d'atributs vàlides per realitzar la firma, estigui fixada al principi del joc de seguretat i per tant la seguretat del nostre esquema és més gran que la dels altres esquemes proposats.

Models de seguretat

La criptografia moderna es basa en problemes suposadament difícils de resoldre amb recursos computacionals limitats. Dit d'una altra manera, s'assumeixen certes hipòtesis sobre problemes difícils, sobre les quals es construeixen els esquemes.

Quan es fa un joc de seguretat i s'analitza la probabilitat d'èxit, el que es fa és relacionar l'èxit del joc amb la solució d'algun problema suposadament difícil. En una demostració de seguretat s'usa la sortida de l'adversari en el joc de seguretat per trencar algun problema difícil. Com que se suposa que el problema és difícil, aleshores també ha de ser difícil que l'adversari tingui èxit en el joc de seguretat i per tant l'esquema serà segur. En altres paraules, es fita superiorment la probabilitat d'èxit de l'adversari per la probabilitat de solucionar un problema computacionalment difícil, que suposem negligible.

Com sabem que un problema és computacionalment difícil? Com sabem que podem assumir alguna hipòtesi sense posar en perill la seguretat dels esquemes criptogràfics si la hipòtesi resulta ser falsa? Els problemes que s'utilitzen són problemes ben coneguts pels matemàtics i presents en la ciència des de fa molts anys. Com que no s'ha trobat cap algorisme eficient per solucionar aquests problemes, s'assumeix que probablement no n'hi ha. Alguns d'aquests problemes són la factorització de grans nombres o el logaritme discret, problemes molt antics dels quals no es coneix cap algorisme amb cost polinòmic per solucionar-los.

Quan reduïm la seguretat d'un esquema a alguna hipòtesi computacional, diem que la seguretat és en el model estàndard, també anomenat model nu. Algunes vegades, però, no n'hi ha prou amb assumir alguna hipòtesi sobre la dificultat de problemes. En aquests casos hem de treballar en altres models de seguretat. Un exemple és el model de l'oracle aleatori [BR93]. En aquest model, en el joc de seguretat el reptador té la capacitat d'assignar un valor de la seva elecció al resultat de fer actuar un hash (una funció que transforma missatges de mida arbitrària a missatges de tamany fix) sobre el missatge. En un escenari real, un hash és una funció determinista i ben definida, de la qual és difícil trobar una anti-imatge donada una imatge. És per això que en el joc de seguretat assumim que podem definir quina és la imatge de cada element. Aquest model s'utilitza sovint quan apareixen funcions de hash en els criptosistemes, però ha estat criticat per diversos criptògrafs, que construeixen esquemes segurs en el model de l'oracle aleatori però que es poden trencar [CGH04].

L'esquema de firma que presentem està basat en el model estàndard, al contrari que altres treballs de firmes basades en atributs. Val a dir, però, que ja existien d'altres esquemes de firma basats en el model estàndard.

Capítol 2

Proves Sense Coneixement

2.1 Introducció

En aquest capítol introduirem el que són les Proves Sense Coneixement (en anglès Zero-Knowledge Proof) [Dam98, Gol00] i explicarem les propietats que han de tenir. Intuïtivament, una Prova Sense Coneixement és un protocol entre un provador i un verificador en la que el provador convenç al verificador de que una sentència és certa. Aquesta prova no revela cap informació al verificador, tret de la veracitat de la sentència. La sentència a que ens referim pot ser, per exemple, que dos grafs són isomorfs o que una equació té alguna solució.

Una Prova Sense Coneixement ha de tenir tres propietats bàsiques.

- La primera és la correcció: si la sentència és veritat i el provador segueix el protocol (direm que és un provador honest), aleshores el verificador acaba acceptant la prova amb probabilitat 1, és a dir, el provador sempre convencerà al verificador.

- La segona és la consistència (en anglès *soundness*): si la sentència és falsa, cap provador que intenti fer trampes pot convèncer al verificador.

- La tercera propietat és el coneixement zero. La idea és que la Prova Sense Coneixement no ha de desprendre cap coneixement més enllà de la correcció de la sentència. Posteriorment veurem com formalitzar aquesta propietat, però la idea és que es pugui simular una Prova Sense Coneixement i aquesta sigui indistingible d'una Prova Sense Coneixement real entre un provador i un verificador.

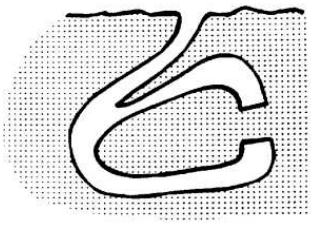
De Proves Sense Coneixement n'hi ha d'interactives i de no interactives. Tot i que en els capítols posteriors només usarem proves no interactives, a continuació presentarem un exemple d'una prova interactiva per ajudar al lector a entendre el concepte.

2.1.1 Com explicar les Proves Sense Coneixement a un nen

En l'any 1990, Quisquater et. al. van publicar un article titulat *How to explain Zero-Knowledge Protocols to Your Children* [QQQ+89] on s'explica una història que il·lustra la idea que hi ha darrera d'una Prova Sense Coneixement.

La història comença a la ciutat de Bagdad, fa molts anys. En aquella ciutat hi vivia Alí Babà. Un dia, quan Alí Babà estava en el mercat, un lladre li va robar la seva bossa. Alí Babà va perseguir el lladre, qui va entrar en una cova que es dividia en dos passadissos: un a l'esquerra i l'altre a la dreta. Alí Babà, que no va veure quin camí prenia el lladre, va decidir anar pel camí de la dreta. A l'arribar al final del camí no va veure ningú. Després va anar cap a l'esquerra, però també es va trobar amb un camí sense sortida on no hi havia ningú.

El dia següent un altre lladre va robar a Alí Babà, qui el va seguir fins a la cova de nou. Tampoc va veure quin camí prenia el lladre, així que va decidir prendre el camí de l'esquerra. Un altre cop, no va veure ningú. Alí Babà va creure que el lladre havia tingut sort, havia anat per l'altre camí i després havia tingut temps a escapar.



A partir d'aquell dia, la mateixa història es va repetir fins a trenta-vuit vegades més. Alí Babà podia pensar que tots els quaranta lladres van tenir sort. Òbviament, això és bastant improbable, així que Alí Babà va pensar que hi havia d'haver un secret amagat. Alí Babà es va quedar amagat darrera d'uns sacs i, després d'una llarga espera, va veure arribar un lladre que, sabent que estava perseguit per la seva víctima, va dir les paraules màgiques 'Obre't, Sèsam'. La paret es va lliscar, connectant els dos camins. El lladre va passar a l'altre camí i la porta es va tancar.

Alí Babà, que havia descobert el secret de la cova misteriosa, va escriure la història en un pergamí. De fet, va aconseguir canviar les paraules màgiques de la mateixa manera que es pot canviar la combinació d'un cadenat, però enlloc d'escriure-les en el pergamí va decidir donar només unes pistes subtils.

El pergamí va arribar a Itàlia a l'edat mitja. Avui en dia es troba als Estats Units, a Boston. Recentment ha atret l'atenció de molts investigadors, i alguns d'ells van aconseguir recuperar les paraules màgiques de les pistes subtils del pergamí. Excavacions arqueològiques van revelar la cova i es va veure que la història del pergamí no era un mite.

Moltes cadenes de televisió es van interessar per aquest fet. Un investigador, Mick Ali,

volia demostrar que coneixia el secret, sense donar-lo a conèixer. Per aconseguir-ho, va fer el següent: primer de tot un equip d'una cadena de televisió va filmar la cova amb els seus dos camins sense sortida. Després tots van sortir de la cova. Mick Ali va entrar per un dels camins, sense que el reporter sabés quin era. Després, un reporter i un càmera van entrar fins a la bifurcació de la cova. El reporter va tirar una moneda enlaire i va decidir que si sortia cara li diria a Mick Ali que tornés pel camí de la dreta i si sortia creu pel camí de l'esquerra. Va sortir cara, així que el reporter va cridar a Mick que sortís per l'esquerra i així ho va fer.

El mateix procediment el van repetir diverses vegades. Van parar a la vegada que feia quaranta, recordant als quaranta lladres. Cada cop que ho feien, Mick Ali anava per un dels dos camins i el reporter el feia tornar pel de la dreta o pel de l'esquerra, segons el que sortís a la moneda. Mick va tenir èxit els quaranta cops.

Una persona que no conegués el secret cada cop hauria pogut fallar l'experiment. La probabilitat que coincideixi el camí triat per la persona amb el resultat del llançament de la moneda els quaranta cops és de $1/2^{40}$. Però el coneixement del secret va permetre a Mick repetir l'experiment amb èxit tants cops com va voler.

Un reporter d'una altra cadena de televisió també volia fer el mateix reportatge. Com que Mick n'havia firmat l'exclusiva, no podia ajudar-lo. Tot i això, li va suggerir que la història es podia gravar sense el coneixement del secret. El reporter va agafar un actor que s'assemblava a Mick Ali i van gravar l'experiment. Com és evident, van haver de repetir moltes de les escenes, ja que l'actor no se'n sortia en la meitat dels casos.

Les dues cadenes de televisió van emetre el reportatge el mateix dia a la mateixa hora. El problema va acabar als jutjats, però el tribunal no va poder decidir quina de les dues històries era verdadera i quina era falsa.

La història simulada no donava cap informació del secret, ja que l'actor no el coneixia pas. Però com que no es podia distingir la història simulada de la real, tampoc se'n desprenia cap secret de la història real. El reporter que va gravar la història verdadera va quedar convençut que Mick Ali tenia el secret, però no va poder usar aquesta informació per convèncer als jutges que la seva història era l'única verdadera.

2.1.2 Comentaris sobre el relat

Aquest relat mostra clarament en què consisteix una prova interactiva amb coneixement zero. En realitat, es tracta d'una prova de coneixement (en anglès, Proof of Knowledge), on a més de demostrar una sentència (la porta s'obre amb certes paraules màgiques) es demostra que es coneix el secret. No aprofundirem en la definició d'una prova de coneixement, però es tracta d'afegir una propietat a les Proves Sense Coneixement de tal manera que es pugui extreure el secret amb el coneixement del provador. Tot i això, he usat l'exemple de la cova misteriosa ja que em sembla força il·lustratiu per explicar com es fa per convèncer a un verificador d'alguna cosa (que es coneix un valor o que una

sentència és certa) i com es fa perquè el verificador no obtingui més informació que allò de què se'l vol convèncer.

Quan vaig llegir el relat, em vaig fer la següent pregunta: per què el reporter no li demanava a Mick que anés per un camí i tornés per l'altre? El relat explicat és fàcil d'entendre gràcies a la seva simplicitat, però hi ha aspectes de les proves que es fan servir en criptografia que no queden reflectits. Si l'experiment fos tal com el que acabo de proposar, queda clar que el segon reporter no l'hauria pogut simular. També és evident que el primer reporter no obtindria cap informació sobre el secret, però en general en la criptografia això no és fàcil d'assegurar. Per garantir que el verificador no obté cap informació sobre el secret s'hauria de comprovar que cap fragment del secret pot ser recuperat i, per tant, s'hauria de fer tota la casuística. Com que això és inviable a la pràctica, es demana que el protocol es pugui simular i amb això es garanteix el fet que el verificador no aprengui res.

2.2 Formalitzant les Proves Sense Coneixement

Un cop vista la intuïció darrera les Proves Sense Coneixement, anem a formalitzar-ne la definició i les propietats. D'ara endavant tractarem només amb proves no interactives i les enfocarem als requisits del següent capítol. Usarem les definicions donades per Groth i Sahai [GS07].

2.2.1 Llenguatges i testimonis

Primer de tot hem de dir quines són les sentències sobre les que fem les demostracions sense coneixement. Un llenguatge L és un conjunt de paraules, és a dir, símbols o cadenes de caràcters. Per exemple, un llenguatge pot ser format per el conjunt de paraules que apareixen en un diccionari. També podem parlar d'un llenguatge en matemàtiques: un llenguatge L pot ser el format per tots els nombres reals. En aquest cas, comprovar que un element pertany a un llenguatge és fàcil. Podem pensar en altres llenguatges en els quals determinar si un element hi pertany no és tant fàcil. Un exemple és el llenguatge format per les equacions que tenen alguna solució entera.

Donada una equació pot ser complicat trobar-ne una solució entera, suposant que n'existeix alguna. Ara bé, *donada* una solució és fàcil comprovar que ho és, només cal substituir els valors en l'equació. A més a més, donada la solució també sabem que l'equació té alguna solució entera i, per tant, pertany al llenguatge de les equacions amb alguna solució entera. La solució donada permet comprovar la pertinença al llenguatge i s'anomena un testimoni (de pertinença al llenguatge).

Anem a formalitzar aquest concepte. Considerem una relació R_L, que pot ser comprovada en temps polinòmic. Donat un element x, diem que w és un testimoni (en anglès witness) si $(x, w) \in R_L$. Definim el llenguatge L com tots aquells elements x que tenen

algun testimoni. Com que la relació és comprovable en temps polinòmic, diem que el llenguatge L és un llenguatge NP.

En l'escenari de les Proves Sense Coneixement, normalment el provador i el verificador hauran de conèixer a priori alguna informació g_k, com ara la descripció d'un grup bilineal. En aquest cas, considerarem una relació avaluable en temps polinòmic i que pot dependre de g_k. Definim el llenguatge L_{g_k} com el conjunt d'elements x que tenen algun testimoni.

2.2.2 Notació

En el següent apartat, assumirem que tots els participants utilitzen algorismes que corren en temps polinòmic, és a dir, executen un nombre d'operacions polinòmic respecte el tamany de les variables que usen com a entrada. Generalment tots tindran com a entrada un paràmetre de seguretat 1^k, que no escriurem explícitament.

Per denotar que un valor c s'obté com a sortida d'un algorisme A amb entrada b i una aleatorietat r, usarem la notació $c \leftarrow A(b)$, on s'entén que s'executa A sobre la distribució de probabilitats uniforme de r (que ometem en la notació). Escrivim $a \leftarrow A; b \leftarrow B(a); \dots$ per denotar un experiment on a s'escull de A, després b s'escull de B i que pot dependre de a i així successivament. Això implica una distribució de probabilitat sobre les sortides, i escrivim $\Pr\Big[a \leftarrow A; b \leftarrow B(a); \cdots : C(a, b, \dots)\Big]$ per la probabilitat de que l'event $C(a, b, \dots)$ es satisfaci després de l'experiment.

Diem que una funció f del paràmetre de seguretat és negligible si per tot $c > 0$ existeix un $K > 0$ tal que per tot $k > K$ tenim $f(k) < k^{-c}$. Denotarem $negl(k)$ per dir que és alguna funció negligible. Si tenim dos funcions f, g escriurem $f(k) \approx g(k)$ si $|f(k) - g(k)| < negl(k)$.

En les proves no interactives que proposem intervenen els següents algorismes (o participants): un algorisme d'inicialització $\mathcal{G}$, que agafa com a entrada únicament el paràmetre de seguretat 1^k i obté com a sortida uns paràmetres (g_k, s_k). En el nostre cas, g_k serà la descripció d'un grup bilinear i s_k serà certa informació secreta, com la factorització de l'ordre del grup o bé serà la cadena de caràcters buida. Considerarem també un algorisme K que genera la cadena de referència comuna σ, que pren com a entrada (g_k, s_k) i treu com a sortida σ. La cadena de referència comuna és una informació pública que, juntament amb el valor de g_k, permetrà definir diferents escenaris. També considerarem un provador P que pren com a entrada (g_k, σ, x, w) i produeix una prova π. Com ja hem comentat, aconseguir un testimoni w pot ser difícil (o no), però en la definició del protocol no ens preocupem com s'obté. Finalment, considerarem un verificador que pren com a entrada (g_k, σ, x, π) i treu 1 si accepta la prova i 0 altrament.

Una entitat que apareix en la definició de les propietats de la prova és el que anomenem un adversari. Aquest adversari modela qualsevol estratègia que intenti trencar l'esquema. L' adversari també fa un nombre d'operacions polinòmic i pot tenir qualssevol entrades i sortides. En algun punt d'aquest capítol, apareixen expressions com $\Pr[(\cdots) : \mathcal{A}(g_k, \sigma) =$

1], on usem el número 1 com a valor booleà. En aquest cas, no ens interessa com $\mathcal{A}$ ha obtingut un 1 com a resultat sinó que ens interessa veure com influeix l'entrada en el resultat, en la probabilitat que la sortida sigui un 1. És a dir, ens interessa saber quina és la probabilitat que $\mathcal{A}$ tregui un 1 amb certes condicions de les variables i comparar-ho amb la probabilitat del mateix succés amb altres condicions. Això ens està donant una mesura de fins quin punt l'adversari és capaç de distingir les condicions sobre les variables d'entrada.

Finalment, en la definició de les propietats també apareix el que anomenem un simulador. Un simulador també fa un nombre d'operacions polinòmic i substitueix a diferents entitats per tenir un comportament diferent. Tot i que aquesta entitat no apareix en cap cas durant el protocol ni modela el comportament de cap participant, ens serveix per demostrar propietats de seguretat. Denotarem per S a un simulador, i si en tenim més ho indicarem amb un subíndex per cada simulador.

2.2.3 Altres consideracions

Com ja s'ha explicat, considerem que totes les entitats executen algorismes amb un nombre polinòmic d'operacions, és a dir, tenen capacitat computacional limitada. En la literatura sovint s'assumeix que el provador té capacitat computacional il·limitada quan es parla de Proves Sense Coneixement. Quan el provador té capacitat computacional limitada, aleshores es parla d'arguments sense coneixement. Per tant, en el nostre cas seria més correcte parlar d'arguments enlloc de proves, però com que és una qüestió menor i en l'article en que basem la secció següent es parla de proves, nosaltres també ho farem.

D'altra banda, quan definim les propietats a vegades direm que una propietat és perfecta. Això fa referència a que la propietat es compleix amb probabilitat 1. No s'ha de confondre amb que sigui una propietat incondicional, doncs això està relacionat amb suposar que l'adversari té capacitat computacional il·limitada, que no és el cas.

2.2.4 Completesa i consistència

Ara ja podem dir que l'objectiu d'una Prova Sense Coneixement és que el provador convenci al verificador que un element x està en un llenguatge L_{g_k} (d'ara endavant l'anomenarem simplement L). Òbviament, x, L i g_k són coneguts tant pel provador com pel verificador.

El provador i el verificador duran a terme un protocol per aconseguir convèncer al segon de que $x \in L$. Les dues primeres propietats que introduirem són la completesa i la consistència. Un protocol no interactiu entre $(\mathcal{G}, K, P, V)$ amb aquestes dues propietats s'anomena un sistema de proves no interactiu.

Completesa perfecta: per tot adversari $\mathcal{A}$ s'ha de complir

$$\Pr\Big[(g_k, s_k) \leftarrow \mathcal{G}; \sigma \leftarrow K(g_k, s_k); (x, w) \leftarrow \mathcal{A}(g_k, \sigma); \pi \leftarrow P(g_k, \sigma, x, w) :$$

$$V(g_k, \sigma, x, \pi) = 1 \text{ si } (g_k, x, w) \in R\Big] = 1$$

Això es llegeix de la següent manera: "la probabilitat que donats uns valors (g_k, s_k) creats per un algorisme $\mathcal{G}$ d'inici; una cadena de referència σ donada per un algorisme K que pren com a entrades (g_k, s_k); una parella (x, w) donat per un algorisme -l'adversari- $\mathcal{A}$, que pren com a entrada (g_k, σ); i una prova π donada per P amb entrades (g_k, σ, x, w); l'algorisme V amb entrades (g_k, σ, x, π) accepti si $(g_k, x, w) \in R$ ha de ser exactament 1".

Aquesta sentència cobreix exactament la intuïció de completesa: sigui quina sigui la parella (x, w) usada pel provador per generar la prova, el verificador queda convençut sempre i quan w sigui un testimoni de que $x \in L$. Aquí s'assumeix que el provador coneixerà un testimoni, cosa que és certa en un provador honest.

També hem dit abans que un provador maliciós no ha de poder enganyar al verificador, és a dir, el verificador no ha d'acceptar la prova si x no és del llenguatge L.

Consistència perfecta: per tot adversari $\mathcal{A}$ s'ha de complir

$$\Pr\Big[(g_k, s_k) \leftarrow \mathcal{G}; \sigma \leftarrow K(g_k, s_k); (x, \pi) \leftarrow \mathcal{A} : V(g_k, \sigma, x, \pi) = 0 \text{ si } x \notin L\Big] = 1$$

Amb aquesta definició queden coberts tots els casos. En aquesta definició l'adversari té èxit si crea una prova vàlida per un $x \notin L$. Aquesta noció es pot generalitzar a una noció de co-consistència: l'adversari té èxit si crea una prova vàlida per $x \in L_{co}$ per algun llenguatge L_{co} que pot dependre de g_k i també de σ. Evidentment, si L_{co} és el complementari de L, aleshores tenim la noció de consistència estàndard.

L_{co}-consistència perfecta:per tot adversari $\mathcal{A}$ s'ha de complir

$$\Pr\Big[(g_k, s_k) \leftarrow \mathcal{G}; \sigma \leftarrow K(g_k, s_k); (x, \pi) \leftarrow \mathcal{A} : V(g_k, \sigma, x, \pi) = 0 \text{ si } x \in L_{co}\Big] = 1$$

2.2.5 Coneixement zero i no distinció de testimoni

Les dues primeres propietats feien referència al que podríem anomenar la correcció de l'esquema: el verificador accepta la prova si, i només si, $x \in L$ (i el provador és honest).

Ara anem a mirar les propietats que tenen relació amb no donar informació al verificador més que la veracitat de la sentència.

Una propietat una mica més relaxada que la de coneixement zero és la de no distinció de testimonis. El que es requereix és que un adversari no sigui capaç de distingir quin testimoni s'ha utilitzat en la construcció de la prova, entre dos d'ells. En el capítol següent s'usa una definició forta d'aquesta propietat: es requereix que un adversari no pugui distingir una cadena de referència comuna real d'una simulada i també es requereix que en la cadena de referència comuna simulada el testimoni usat pel provador sigui perfectament indistingible. Un sistema de proves no interactiu amb aquesta propietat s'anomena un sistema de proves no interactiu amb no distinció de testimoni.

No distinció de testimoni: per tot adversari $\mathcal{A}$ s'ha de complir

$$\Pr\Big[(g_k, s_k) \leftarrow \mathcal{G}; \sigma \leftarrow S(g_k, s_k); (x, w_0, w_1) \leftarrow \mathcal{A}(g_k, \sigma); \pi \leftarrow P(g_k, \sigma, x, w_0) : \mathcal{A}(\pi) = 1\Big]$$
$$= \Pr\Big[(g_k, s_k) \leftarrow \mathcal{G}; \sigma \leftarrow S(g_k, s_k); (x, w_0, w_1) \leftarrow \mathcal{A}(g_k, \sigma); \pi \leftarrow P(g_k, \sigma, x, w_1) : \mathcal{A}(\pi) = 1\Big]$$

i

$$\Pr\Big[(g_k, s_k) \leftarrow \mathcal{G}; \sigma \leftarrow K(g_k, s_k) : \mathcal{A}(g_k, \sigma) = 1\Big]$$
$$\approx \Pr\Big[(g_k, s_k) \leftarrow \mathcal{G}; \sigma \leftarrow S(g_k, s_k) : \mathcal{A}(g_k, \sigma) = 1\Big]$$

En el capítol següent construirem proves amb la propietat de no distinció de testimoni i a partir d'aquestes construirem Proves Sense Coneixement.

Ja hem dit com introduir la propietat de coneixement zero: s'ha de poder simular una prova, sense conèixer cap testimoni, indistingible d'una feta per un provador. Evidentment, amb la informació pública això no té perquè ser possible i per això se li dona al simulador certa informació τ, que permet fer trampes. Aquesta informació no està relacionada amb cap testimoni sinó amb els paràmetres secrets d'inici. En el nostre cas, es genera en el moment de crear la cadena de referència comuna. A l'adversari també se li dona la informació secreta. Un sistema de proves no interactiu amb aquesta propietat s'anomena un sistema de proves no interactiu amb coneixement zero i una prova en aquest sistema s'anomena una Prova Sense Coneixement.

Tot i que hi ha definicions de la propietat de coneixement zero més relaxades, l'esquema del capítol següent compleix la definició que ara donarem.

Coneixement zero: per tot adversari $\mathcal{A}$ s'ha de complir

$$\Pr\Big[(g_k, s_k) \leftarrow \mathcal{G}; \sigma \leftarrow K(g_k, s_k) : \mathcal{A}(g_k, \sigma) = 1\Big]$$
$$\approx \Pr\Big[(g_k, s_k) \leftarrow \mathcal{G}; (\sigma, \tau) \leftarrow S_1(g_k, s_k) : \mathcal{A}(g_k, \sigma) = 1\Big]$$

i

$$\Pr\Big[(g_k, s_k) \leftarrow \mathcal{G}; (\sigma, \tau) \leftarrow S_1(g_k, s_k); (x, w) \leftarrow \mathcal{A}(g_k, \sigma, \tau); \pi \leftarrow P(g_k, \sigma, x, w) : \mathcal{A}(\pi) = 1\Big]$$
$$= \Pr\Big[(g_k, s_k) \leftarrow \mathcal{G}; (\sigma, \tau) \leftarrow S_1(g_k, s_k); (x, w) \leftarrow \mathcal{A}(g_k, \sigma, \tau); \pi \leftarrow S_2(g_k, \sigma, \tau, w) : \mathcal{A}(\pi) = 1\Big]$$

La diferència entre una prova amb no distinció de testimoni i una Prova Sense Coneixement és subtil. És evident que la segona propietat és més forta que la primera: si no s'obté cap informació en particular és impossible distingir el testimoni usat. Ara bé, la propietat de no distinció de testimoni és més feble que la de coneixement zero. Una prova amb no distinció de testimoni pot donar certa informació al verificador, com per exemple informació sobre el conjunt de testimonis possibles pel valor x. En particular, si només hi ha un testimoni per x no està garantit que l'adversari no pugui obternir el testimoni de la prova.

2.3 Compromisos

En aquesta secció explicarem què són els compromisos [Dam98] i quines propietats han de tenir. Intuïtivament, comprometre un valor vol dir calcular un altre valor que està lligat de manera única al valor compromès de manera que al mateix temps sigui difícil esbrinar quin valor està compromès.

Un primer ús dels compromisos és solucionar problemes de computació amb diversos participants. Per exemple, suposem que dues persones estan a certa distància i, comunicant-se per internet, volen obtenir un valor binari aleatori comú de manera que cap d'ells pugui fer trampes. Una solució és que cadascú tiri una moneda, comprometi el resultat i enviï el compromís a l'altra persona. Un cop cadascú ha obtingut el compromís de l'altra persona, es poden *obrir* els compromisos i calcular la suma (XOR) de les dues monedes. En aquest cas, obrir un compromís consisteix en revelar la informació necessària per tal de recuperar el valor compromès.

Però no sempre és necessari obrir un compromís. En les Proves Sense Coneixement, interessarà construir compromisos per amagar certa informació. Gràcies a que el compromís està lligat amb un valor determinat, el verificador pot confiar que dins el compromís hi ha el valor amagat i que aquest no canviarà en cap moment.

A continuació descrivim les propietats que ha de tenir un compromís. De la mateixa manera que en els sistemes de proves, suposarem que hi ha un algorisme d'inicialització $\mathcal{G}$, que agafa com a entrada el paràmetre de seguretat 1^k i obté com a sortida uns paràmetres (g_k, s_k). Considerarem també un algorisme K que genera la cadena de referència comuna σ, que pren com a entrada g_k, s_k i treu com a sortida σ. Suposem que hi ha un algorisme de compromís, *com*, que pren com a entrada (g_k, σ, x) i dona a la sortida un valor c. Finalment considerem un adversari $\mathcal{A}$.

Un sistema de compromís ha de tenir la propietat d'amagar i de lligar.

Amagar: per tot adversari $\mathcal{A}$ s'ha de complir

$$\begin{aligned}&\Pr\Big[(g_k, s_k) \leftarrow \mathcal{G}; \sigma \leftarrow K(g_k, s_k); (m_0, m_1) \leftarrow \mathcal{A}(g_k, \sigma); c \leftarrow com(g_k, \sigma, m_0) : \mathcal{A}(c) = 1\Big] \\ = &\Pr\Big[(g_k, s_k) \leftarrow \mathcal{G}; \sigma \leftarrow K(g_k, s_k); (m_0, m_1) \leftarrow \mathcal{A}(g_k, \sigma); c \leftarrow com(g_k, \sigma, m_1) : \mathcal{A}(c) = 1\Big]\end{aligned}$$

La propietat de lligar demana que existeixi una aplicació *dec* que permeti obrir els compromisos. Aquesta aplicació rep com a entrada el compromís i certa aleatorietat i obté com a sortida un valor identificat amb el valor compromès. El que diu la propietat és que la probabilitat d'obrir un compromís a dos valors diferents ha de ser aproximadament zero. No exigim la igualtat ja que en algun esquema de compromís del capítol següent la probabilitat no serà zero, però serà negligible.

Lligar: per tot adversari $\mathcal{A}$ s'ha de complir

$$\Pr\Big[(g_k, s_k) \leftarrow \mathcal{G}; \sigma \leftarrow K(g_k, s_k); (c, r_1, r_2) \leftarrow \mathcal{A}(g_k, \sigma); m_1 \leftarrow dec(g_k, \sigma, c, r_1);$$

$$m_2 \leftarrow dec(g_k, \sigma, c, r_2) : m_1 \neq m_2 \Big] \approx 0$$

Com es pot veure, les propietats d'amagar i de lligar estan molt relacionades amb la de no distinció de testimoni i consistència, respectivament, de les Proves Sense Coneixement. De fet, en el capítol següent usarem compromisos amb una de les dues qualitats i després ens basarem en problemes d'indecisió per tal d'aconseguir la no distinció de testimoni i la consistència al mateix temps.

Capítol 3

Proves Sense Coneixement de Groth-Sahai

3.1 Introducció

En la literatura hi ha hagut diferents propostes per realitzar Proves Sense Coneixement no interactives (d'ara endavant proves NIZK, de l'anglès Non-Interactive Zero-Knowledge proofs). El primer resultat important, [BFM88], demostra que es poden construir proves NIZK per qualsevol llenguatge NP. En aquest article es construeixen proves per qualsevol llenguatge NP, però són altament ineficients. Aquesta ineficiència ve donada per dos motius. El primer és que per realitzar les proves primer s'ha de fer una reducció NP molt costosa, com ara una reducció a acompliment de circuits. El segon és que les proves donades per l'acompliment de circuits són molt ineficients, per molt petits que siguin aquests circuits.

El segon problema s'aconsegueix resoldre en [GOS06], on es proposa un nou sistema de proves NIZK per l'acompliment de circuits. En l'article s'explica com usar pairings bilineals per aconseguir proves eficients, però tot i això encara s'ha de realitzar la costosa reducció NP.

Posteriorment hi ha diversos articles que intenten solucionar el problema de la reducció NP. Per una banda, [BW06, BW07] proposen proves NIZK per problemes relacionats amb firma digital, obtenint proves per sentències molt concretes. Per altra banda, [Gro06] és el primer en proposar un llenguatge depenent del grup i proves NIZK per afirmacions en aquest llenguatge. En l'article es construeixen proves per l'acompliment d'equacions on hi intervenen pairings. El problema, és que les proves són molt grans, inclús per equacions petites.

Finalment, Groth i Sahai [GS07] construeixen un sistema de proves NIZK que permet demostrar l'acompliment de conjunts d'equacions algebraiques que surten en grups bilineals. Les afirmacions que permet provar són molt més generals que ens esquemes

anteriors, mantenint l'eficiència de les proves.

Cal remarcar que ens hem centrat en Proves Sense Coneixement no interactives, si mirem les Proves Sense Coneixement interactives l'escenari és ben diferent, tot i que no hi ha cap sistema que permeti fer proves sobre afirmacions tant generals com l'esquema de Groth-Sahai.

En aquest capítol explicarem a fons les proves NIZK de Groth i Sahai [GS07], doncs les usarem per construir l'esquema de firma que es presenta posteriorment.

3.2 Equacions sobre grups amb aplicació bilineal

Com ja hem comentat, les proves de Groth-Sahai permeten demostrar en coneixement zero i sense interacció l'acompliment d'equacions que apareixen en grups bilineals, amb proves eficients. Com que la construcció de les Proves Sense Coneixement es basa en proves amb la propietat de no distinció de testimoni, primer explicarem com construir les segones i posteriorment construirem les Proves Sense Coneixement. A continuació descrivim els tipus d'equació que permeten fer-ne proves amb la propietat de no distinció de testimoni, d'ara endavant NIWI (de l'anglès, Non-Interactive Witness-Indistinguishable).

En tot moment, usarem la següent notació: suposem que tenim un grup bilineal, definit en la secció 1. Denotem per X_i les variables en $\mathbb{G}_1$, Y_i les variables en $\mathbb{G}_2$, x_i i y_i les variables en $\mathbb{Z}_n$. També usem g_i i h_i per denotar constants en $\mathbb{G}_1$ i $\mathbb{G}_2$ respectivament i a_i, b_i i $\gamma_{i,j}$ per denotar constants en $\mathbb{Z}_n$. Finalment, t_T denota una constant de $\mathbb{G}$, t_1 i t_2 denoten constants en $\mathbb{G}_1$ i $\mathbb{G}_2$ respectivament i t denota una constant en $\mathbb{Z}_n$.

Equacions de producte de pairings:

$$\prod_{i=1}^{n} e(g_i, Y_i) \cdot \prod_{i=1}^{m} e(X_i, h_i) \cdot \prod_{i=1}^{m}\prod_{j=1}^{n} e(X_i, Y_j)^{\gamma_{ij}} = t_T$$

Equacions multi-escalars en $\mathcal{G}_1$:

$$\prod_{i=1}^{n'} g_i^{y_i} \cdot \prod_{i=1}^{m} X_i^{b_i} \cdot \prod_{i=1}^{m}\prod_{j=1}^{n'} X_i^{\gamma_{ij} y_j} = t_1$$

Equacions multi-escalars en $\mathcal{G}_2$:

$$\prod_{i=1}^{n} Y_i^{a_i} \cdot \prod_{i=1}^{m'} h_i^{x_i} \cdot \prod_{i=1}^{m'}\prod_{j=1}^{n} Y_j^{\gamma_{ij} x_j} = t_2$$

Equacions quadràtiques en $\mathbb{Z}_n$:

$$\sum_{i=1}^{n'} a_i y_i + \sum_{i=1}^{m'} x_i b_i + \sum_{i=1}^{m'}\sum_{j=1}^{n'} \gamma_{ij} x_i y_j = t$$

Com entendrem després, ens hem de reduir a aquests tipus d'equacions, que en el cas de $\mathbb{Z}_n$ són quadràtiques i en altres casos les variables van, com a molt, de dos en dos (en aquest sentit també són equacions quadràtiques).

Com hem dit abans, aquestes equacions permeten fer proves NIWI d'una gran quantitat d'afirmacions. Per posar un exemple, per demostrar que $h_1 = g^x$ i $h_2 = g_2^x$, és a dir, que h_1 i h_2 tenen el mateix logaritme discret amb base g_1 i g_2 respectivament, n'hi ha prou amb veure que

$$e(h_2, g_1) \cdot e(h_1, g_2^{-1}) = 1$$

3.3 Visió general de la metodologia de Groth-Sahai per fer proves

Un cop hem vist les equacions que permeten fer-ne proves, anem a veure com fer-les exactament. El procediment serà fer compromisos de les variables i donar certa informació addicional per convèncer al verificador que les variables compromeses compleixen un conjunt d'equacions. És a dir, volem convèncer al verificador que els compromisos contenen variables que compleixen un conjunt d'equacions.

La idea general és que interpretarem $\mathbb{G}_1, \mathbb{G}_2$ i $\mathbb{G}_T$ com a mòduls sobre $\mathbb{Z}_n$ (recordem que un mòdul és l'extensió del concepte d'espai vectorial quan no tenim un cos sinò un anell). També veurem $\mathbb{Z}_n$ com un $\mathbb{Z}_n$-mòdul. En una equació tindrem variables i constants de tres mòduls que anomenarem $\mathbb{A}_1$,$\mathbb{A}_2$ i $\mathbb{A}_T$. Per exemple, en les equacions de producte de pairings, definim $\mathbb{A}_1 = \mathbb{G}_1$, $\mathbb{A}_2 = \mathbb{G}_2$ i $\mathbb{A}_T = \mathbb{G}_T$ i en les equacions multi-escalars en $\mathbb{G}_1$ tindrem $\mathbb{A}_1 = \mathbb{G}_1$, $\mathbb{A}_2 = \mathbb{Z}_n$ i $\mathbb{A}_T = \mathbb{G}_1$.

Aquesta visió de $\mathbb{Z}_n$-mòduls ens permetrà definir uns altres mòduls $\mathbb{B}_1, \mathbb{B}_2$ i $\mathbb{B}_T$, amb una aplicació bilineal de $\mathbb{B}_1 \times \mathbb{B}_2$ a $\mathbb{B}_T$ equivalent al pairing que tenim en el primer tipus d'equacions, a una exponenciació en $\mathbb{G}_1$ o $\mathbb{G}_2$, o al producte de dos nombres de $\mathbb{Z}_n$, que també són aplicacions bilineals. Definirem també aplicacions que ens permeten associar elements de $\mathbb{A}_1, \mathbb{A}_2, \mathbb{A}_T$ a elements de $\mathbb{B}_1, \mathbb{B}_2, \mathbb{B}_T$ respectivament. Gràcies al fet que totes les aplicacions seran lineals i a la manera com les definirem, podrem traslladar equacions amb constants i variables en $\mathbb{A}_1, \mathbb{A}_2, \mathbb{A}_T$ a equacions en $\mathbb{B}_1, \mathbb{B}_2, \mathbb{B}_T$.

$$\begin{array}{ccccc} \mathbb{A}_1 & \times & \mathbb{A}_2 & \xrightarrow[f]{} & \mathbb{A}_T \\ \iota_1 \downarrow\uparrow p_1 & & \iota_2 \downarrow\uparrow p_2 & & \iota_T \downarrow\uparrow p_T \\ \mathbb{B}_1 & \times & \mathbb{B}_2 & \xrightarrow[F]{} & \mathbb{B}_T \end{array}$$

Figura 3.1: *Diagrama de mòduls i aplicacions*

El fet de treballar en uns mòduls $\mathbb{B}_1, \mathbb{B}_2, \mathbb{B}_T$, que normalment seran més grans que els mòduls de partida $(\mathbb{A}_1, \mathbb{A}_2, \mathbb{A}_T)$, ens permetrà fer els compromisos. Primer, definim dos tipus de *claus de compromís* diferents. Una clau del primer tipus permetrà garantir que el compromís està unívocament lligat al valor compromès i una clau del segon tipus permetrà garantir que, donat un compromís, el valor compromès pot ser qualsevol. És a dir, si fem un compromís amb la primera clau tindrem la propietat de lligar (incondicional) i amb la segona tindrem la propietat d'amagar (incondicional).

Està clar que un compromís 'ideal' és aquell que té les dues propietats però això és impossible degut a la definició de cada propietat. El que sí que és possible és tenir una propietat incondicional i l'altra propietat computacional. El punt clau de les proves de Groth-Sahai és que els mòduls són tals que podem usar certs problemes de decisió, és a dir, que no podrem distingir computacionalment quines característiques tenen certs valors. Això vol dir que no podrem distingir (des del punt de vista computacional) si una clau de compromís és del primer tipus o del segon. D'aquesta manera, si la clau que s'usa té la propietat de lligar (incondicional) també tindrà la propietat d'amagar (computacionalment). Això és degut a que si algú pogués distingir el valor que s'ha compromès, aleshores estaria esbrinant que la clau és del primer tipus i, per tant, estaria trencant el problema de decisió, que estem suposant difícil. Amb això hem aconseguit compromisos amb una propietat incondicional i la segona propietat computacional, que és el màxim a que podem aspirar. El fet d'usar una clau o l'altra dependrà de l'escenari concret.

Finalment, com que en comprometre les variables estem afegint informació per amagar el valor, l'equació original (un cop traslladada als mòduls $\mathbb{B}_1, \mathbb{B}_2, \mathbb{B}_T$) ja no es complirà. Farà falta afegir la informació sobrant, que anomenarem pròpiament prova NIWI o simplement prova.

La resta d'aquest capítol anirà organitzat de la següent manera: en la secció 3.4 s'exposen les diferents hipòtesis de decisió en què es basen les particularitzacions proposades per Groth i Sahai. En la secció 3.5 s'explica la notació que s'usarà per fer la construcció el més general possible. En la secció 3.6 s'explica com comprometre un valor. En la secció 3.7 s'explica com escollir tots els paràmetres necessaris per fer les proves NIWI. En la secció 3.8 s'explica com calcular els valors que configuren les proves NIWI. En la secció 3.9 es fa un resum de tots els algorismes per generar i comprovar proves NIWI i finalment en la secció 3.10 es fa un esbós de com construir Proves Sense Coneixement.

3.4 Hipòtesis de decisió

La metodologia de Groth-Sahai per construir proves NIWI es planteja sobre mòduls en general, el que permet molta flexibilitat en el moment de particularitzar aquesta metodologia. En l'article es proposen tres particularitzacions basades en problemes de decisió comuns en la literatura. Aquests problemes són: el problema de decisió de subgrup, el

problema Diffie-Hellman extern simètric (SXDH, de l'anglès, Symmetric eXternal Diffie-Hellman) i el problema decisional lineal (DLIN, de l'anglès, Decisional LINear).

Decisió de subgrup: La primera particularització es basa en grups d'ordre compost, introduïts per Boneh, Goh i Nissim [BGN05]. Partim doncs d'un grup bilineal d'ordre compost $(n, \mathbb{G}, \mathbb{G}_T, e, g)$ on $n = pq$. Podem escriure $\mathbb{G}$ com $\mathbb{G}_p \times \mathbb{G}_q$, on $\mathbb{G}_p, \mathbb{G}_q$ són els subgrups d'ordre primer p i q respectivament. La hipòtesis de la decisió de subgrup diu que és difícil distingir un element aleatori en $\mathbb{G}$ d'un element aleatori en $\mathbb{G}_q$.

SXDH: Sigui $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g_1, g_2)$ un grup bilineal d'ordre primer. La hipòtesis del Diffie-Hellman extern diu que el problema decisional Diffie-Hellman és difícil en un dels grups $\mathbb{G}_1$ o $\mathbb{G}_2$ [BBS04]. S'anomena Diffie-Hellman extern per diferenciar aquesta hipòtesis de la hipòtesis bilineal decisional Diffie-Hellman, que amb l'existència d'un pairing és una hipòtesi incorrecta ja que el problema decisional resulta ser fàcil amb l'ús del pairing. La hipòtesi Diffie-Hellman extern simètric diu que el problema decisional Diffie-Hellman és difícil tant en $\mathbb{G}_1$ com en $\mathbb{G}_2$.

DLIN: La hipòtesi decisional lineal per un grup bilineal d'ordre primer $(p, \mathbb{G}, \mathbb{G}_T, e, g)$ introduïda per Boneh, Boyen i Sacham [BBS04] diu que donat $(g^\alpha, g^\beta, g^{r\alpha}, g^{s\beta}, g^t)$ és difícil distingir si $t = r + s$ o si t és un element aleatori.

3.5 Unificant i simplificant la notació

En l'article de Groth i Sahai [GS07], proposen un marc teòric general, independent del tipus d'equació a tractar, per construir proves NIWI. En aquest marc general s'usa la notació additiva per simplicitat. S'ha de tenir en compte, però, que en les particularitzacions s'usarà notació additiva o multiplicativa depenent de l'equació a tractar. Anem a veure com Groth i Sahai unifiquen la notació en el seu article.

Prenem un anell $\mathcal{R}$ i $\mathbb{A}_1, \mathbb{A}_2, \mathbb{A}_T$ uns $\mathcal{R}$-mòduls amb una aplicació bilineal $f : \mathbb{A}_1 \times \mathbb{A}_2 \to \mathbb{A}_T$. Considerarem equacions quadràtiques en les variables $x_1, \ldots, x_m \in \mathbb{A}_1, y_1, \ldots, y_n \in \mathbb{A}_2$ de la forma

$$\sum_{j=1}^{n} f(a_j, y_j) + \sum_{i=1}^{m} f(x_i, b_i) + \sum_{i=1}^{m}\sum_{j=1}^{n} \gamma_{ij} f(x_i, y_j) = t$$

Per simplificar la notació, si $\vec{x} = (x_1, \ldots, x_n) \in \mathbb{A}_1$ i $\vec{y} = (y_1, \ldots, y_n) \in \mathbb{A}_2$, definim

$$\vec{x} \cdot \vec{y} = \sum_{i=1}^{n} f(x_i, y_i)$$

Totes les equacions es poden escriure com

$$\vec{a} \cdot \vec{y} + \vec{x} \cdot \vec{b} + \vec{x} \cdot \Gamma \vec{y} = t$$

És important remarcar que, degut a la propietat de bilinearitat de f, donada qualsevol matriu $\Gamma \in \mathrm{Mat}_{n\times n}(\mathcal{R})$ i qualssevol $\vec{x} \in \mathbb{A}_1^m, \vec{y} \in \mathbb{A}_2^n$ tenim que $\vec{x} \cdot \Gamma\vec{y} = \Gamma^\top \vec{x} \cdot \vec{y}$. Aquesta propietat es farà servir al llarg de l'explicació.

Anem a veure com particularitzar aquesta simplificació per cada tipus d'equacions de les descrites en el capítol anterior, tenint en compte si usem notació additiva o multiplicativa. En notació multiplicativa tindrem $\vec{x} \cdot \vec{y} = \prod_{i=1}^{n} f(x_i, y_i)$, òbviament.

Equacions de producte de pairings:

Definim $\mathcal{R} = \mathbb{Z}_n, \mathbb{A}_1 = \mathbb{G}_1, \mathbb{A}_2 = \mathbb{G}_2, \mathbb{A}_T = \mathbb{G}_T, f(x,y) = e(x,y)$ i $\vec{x} \cdot \vec{y} = \prod_{i=1}^{n} e(x_i, y_i)$

Equacions multi-escalars en $\mathcal{G}_1$:

Definim $\mathcal{R} = \mathbb{Z}_n, \mathbb{A}_1 = \mathbb{G}_1, \mathbb{A}_2 = \mathbb{Z}_n, \mathbb{A}_T = \mathbb{G}_1, f(X,y) = X^y$ i $\vec{X} \cdot \vec{y} = \prod_{i=1}^{n} X_i^{y_i}$

Equacions multi-escalars en $\mathcal{G}_2$:

Definim $\mathcal{R} = \mathbb{Z}_n, \mathbb{A}_1 = \mathbb{Z}_n, \mathbb{A}_2 = \mathbb{G}_2, \mathbb{A}_T = \mathbb{G}_2, f(x,Y) = Y^x$ i $\vec{x} \cdot \vec{Y} = \prod_{i=1}^{n} Y_i^{x_i}$

Equacions quadràtiques en $\mathbb{Z}_n$:

Definim $\mathcal{R} = \mathbb{Z}_n, \mathbb{A}_1 = \mathbb{Z}_n, \mathbb{A}_2 = \mathbb{Z}_n, \mathbb{A}_T = \mathbb{Z}_n, f(x,y) = xy \mod n$ i $\vec{x} \cdot \vec{y} = \sum_{i=1}^{n} x_i y_i$

3.6 Realitzant els compromisos

Com ja hem introduït, per realitzar una prova NIWI primer es fan compromisos de les variables $x_1, \ldots, x_m \in \mathbb{A}_1, y_1, \ldots, y_n \in \mathbb{A}_2$. Això es fa assignant-los valors d'uns altres $\mathcal{R}$-mòduls $\mathbb{B}_1, \mathbb{B}_2$ i fent els compromisos en aquests mòduls.

En aquest capítol ens centrarem a comprometre valors d'un sol $\mathcal{R}$-mòdul $\mathbb{A}$. Per comprometre valors de $\mathbb{A}$ necessitarem certs paràmetres, el que anomenarem la cadena comuna de referència (d'ara endavant CRS, de l'anglès Common Reference String). El CRS descriurà un altre $\mathcal{R}$-mòdul $\mathbb{B}$, i aplicacions $\mathcal{R}$-lineals $\iota : \mathbb{A} \to \mathbb{B}$, la inclusió, i $p : \mathbb{B} \to \mathbb{A}$, la projecció. A més a més, la clau pública ha de contenir certs elements $u_1, \ldots, u_{\hat{m}} \in \mathbb{B}$. Per comprometre $x \in \mathbb{A}$ escollim elements aleatoris $r_1, \ldots, r_{\hat{m}} \in \mathcal{R}$ i calculem el valor

$$c = \iota(x) + \sum_{i=1}^{\hat{m}} r_i u_i$$

Com ja hem comentat, l'esquema de compromisos té dos tipus de claus:

Clau d'amagar: Una clau d'amagar conté $(\mathbb{B}, \iota, u_1, \ldots, u_{\hat{m}})$ tal que $\iota(\mathbb{A}) \subseteq \langle u_1, \ldots, u_{\hat{m}} \rangle$[1]. El compromís $c = \iota(x) + \sum_{i=1}^{\hat{m}} r_i u_i$ amaga perfectament quan $r_1, \ldots, u_{\hat{m}}$ són elements aleatoris de $\mathcal{R}$. Això és conseqüència de que per tota parella $c \in \mathbb{B}, x \in \mathbb{A}$ existeixen valors $r_1, \ldots, u_{\hat{m}}$ tals que $c = \iota(x) + \sum_{i=1}^{\hat{m}} r_i u_i$, o el que és el mateix, un valor c pot comprometre qualsevol valor.

Clau de lligar: Una clau de lligar conté $(\mathbb{B}, \iota, p, u_1, \ldots, u_{\hat{m}})$ tal que $\forall i$ tenim $p(u_i) = 0$ (en notació multiplicativa, $p(u_i) = 1$) i $p \circ \iota$ no és eficientment computable. El compromís $c = \iota(x) + \sum_{i=1}^{\hat{m}} r_i u_i$ conté per tant la informació no trivial $p(c) = p(\iota(x))$ sobre x. En el cas particular que $p \circ \iota$ sigui l'identitat, aleshores el compromís lliga perfectament.

Abans de passar al següent punt m'agradaria aprofundir una mica més sobre l'aplicació p. Primer de tot, cal tenir present que aquesta aplicació només té sentit quan usem una clau de lligar, amb una clau d'amagar aquesta aplicació no té sentit ja que no es pot definir cap subespai on fer la projecció que elimini la aleatorietat. Pel que fa a la computabilitat de p també és necessari fer uns aclariments. És clar que si qualsevol persona pot calcular p, aleshores es trenca la idea d'amagar un valor i el compromís no serveix per res. Al llarg de les particularitzacions apareixeran diverses maneres de definir p. Utilitzant un esquema de xifrat aconseguim que p sigui difícil de calcular *si no es coneix la informació necessària*: només qui conegui la clau secreta podrà calcular l'aplicació p, que és l'aplicació de desxifrar. Això té l'avantatge que, per exemple, una entitat rastrejadora podrà obrir el compromís en cas de frau. També es poden usar mètodes de computació distribuïda de manera que ningú conegui la clau secreta que permet calcular p. Una altra possibilitat és que p sigui l'inversa d'una funció unidireccional i per tant sigui difícilment computable. Un exemple és el cas en que p és el logaritme discret. En tot cas, la idea és que encara que no es pot calcular p, el verificador pot estar segur que el compromís i el valor compromès estan lligats unívocament (o gairebé, segons qui sigui $p \circ \iota$) ja que si algú tingués capacitat de càlcul il·limitada, aleshores podria trobar aquesta relació.

Impossibilitat de distingir computacionalment: amb els dos tipus de claus apareix la següent qüestió: si usem una clau d'amagar com assegurem que no es pot fer trampes comprometent un valor i després dient que el valor és un altre? Hem dit que un compromís c pot comprometre a qualsevol x, cosa que sembla permetre fer trampes. De la mateixa manera, si usem una clau de lligar, com assegurem que del compromís no se'n desprén cap informació sobre el valor compromès x? Com ja s'ha comentat abans, això s'aconsegueix treballant en mòduls on assumim certes hipòtesis de decisió. Com que no podrem distingir una clau d'amagar d'una clau de lligar, tindrem les propietats d'amagar i lligar alhora, amb una de les dues propietats incondicional i l'altra propietat computacional. Si un adversari pot esbrinar informació sobre el valor compromès x, aleshores està esbrinant de quin tipus és la clau (amb una clau d'amagar és impossible obtenir informació

[1] D'ara endavant farem servir $\langle u_1, \ldots, u_{\hat{m}} \rangle$ tant en notació additiva com multiplicativa, com a subespai generat additivament o multiplicativament per aquests valors

de x) i per tant està trencant el problema difícil. El mateix argument aplica al cas que un adversari comprometi un valor x i després pretengui convèncer a algú de que el valor compromés és un altre.

Com que serà d'interès comprometre diverses variables a la vegada, definirem certa notació. Donats elements $x_1, \ldots, x_m \in \mathbb{A}$ escriurem $\vec{c} = \iota(\vec{x}) + R\vec{u}$ amb $R \in \mathrm{Mat}_{m \times \hat{m}}(\mathcal{R})$ per fer compromisos $c_1, \ldots, c_m$ calculats com $c_i = \sum_{j=1}^{\hat{m}} r_{ij} u_j$. Aquesta expressió només té sentit quan utilitzem notació additiva. El que farem és definir una notació per poder-ho traslladar a notació multiplicativa.

D'ara endavant, si tenim $\vec{a} \in \mathcal{R}^n$ i $\vec{x} \in \mathbb{B}^n$, usarem

$$exp_v(\vec{a}, \vec{x}) = x_1^{a_1} \odot x_2^{a_2} \odot \cdots \odot x_n^{a_n}$$

on $\odot$ és la multiplicació d'elements d'un mòdul - si el mòdul és un grup, és la multiplicació del grup, altrament ja ho definirem. Finalment, si $H \in \mathrm{Mat}_{m \times n}(\mathcal{R})$, definim

$$exp_M(H, \vec{x}) = (exp_v(H^{(1)}, \vec{x}), \exp_v(H^{(2)}, \vec{x}), \ldots, \exp_v(H^{(m)}, \vec{x})),$$

on $H^{(m)}$ és la fila i-éssima de la matriu H.

Tot i que pot semblar una notació estranya i confusa, en el fons és una generalització dels conceptes *multiplicar un vector per un altre* i *multiplicar una matriu per un vector*, tenint en compte que estem usant notació multiplicativa enlloc de notació additiva.

3.6.1 Particularitzacions

El tractament dels compromisos usant el llenguatge dels mòduls generalitza alguns dels resultats existents en compromisos sobre els grups bilineals, per exemple [BGN05, GOS06, Gro06].

A continuació, particularitzarem els compromisos pels tres escenaris explicats en la secció 3.4. Distingim dos tipus d'elements a comprometre: podem comprometre elements del grup $\mathbb{G}_1$ o $\mathbb{G}_2$ o bé podem comprometre exponents, és a dir, elements de $\mathbb{Z}_n$. En el moment de comprometre exponents ens interessarà usar les mateixes claus que quan es compromet elements del grup i reduir el nombre de claus necessàries per comprometre sempre que sigui possible. Com ara veurem, al comprometre exponents anirem a parar al mateix mòdul que si comprometem elements del grup. El motiu l'introduirem més endavant, però la idea és que una mateixa variable pot aparèixer en més d'un tipus d'equació i per tant voldrem fer les equacions en els mòduls $\mathbb{B}_1, \mathbb{B}_2$ i $\mathbb{B}_T$ el més homogènies possible.

Decisió de subgrup. En aquesta particularització tenim un grup $\mathbb{G}$ d'ordre compost $n = pq$ amb un generador g. Aquest grup es pot veure com un $\mathbb{Z}_n$-mòdul, usant la notació anterior definim $\mathbb{A} = \mathbb{G}$ i $\mathbb{B} = \mathbb{G}$. La clau de compromís contindrà un element $h \in \mathbb{G}$. Podem escollir h tal que h generi $\mathbb{G}$ o bé h tingui ordre q. La hipòtesis de la decisió de subgrup ens diu que aquests dos tipus de claus són indistingibles.

Sigui $\iota : \mathbb{G} \to \mathbb{G}$ l'aplicació identitat. Si h genera $\mathbb{G}$, aleshores $c = \iota(X)h^r$ amaga perfectament. Per altra banda, si h té ordre q definim $\delta_p \in \mathbb{Z}_n$ tal que $\delta_p = 1 \mod p$ i $\delta_p = 0 \mod q$ i també definim l'aplicació $p : \mathbb{G} \to \mathbb{G}$, que assigna a elements en el subgrup de $\mathbb{G}$ d'ordre p calculant $p(X) = X^{\delta_p}$. Està clar que $p(h) = 1$, recordem que estem usant notació multiplicativa. Aquesta aplicació defineix X únicament en $\mathbb{G}_p$. Amb això no aconseguim un lligam incondicional, doncs si X i Y són elements de $\mathbb{G}$ diferents però projectats en $\mathbb{G}_p$ són el mateix element, aleshores $p(\iota(X)) = p(\iota(Y))$. Tot i això, la probabilitat que donats $X, Y \in \mathbb{G}$ a l'atzar succeeixi això és de $1/q$. Com que estem suposant que n és difícil de factoritzar, aleshores q és prou gran (1024 bits aproximadament) i per tant la probabilitat de trobar aquests X i Y és molt petita. En aquest cas, però, enlloc de consistència tindrem L_{co}-consistència, definida en el capítol 2.

Si volem comprometre els exponents, usem els mòduls $\mathbb{A}' = \mathbb{Z}_n$ i $\mathbb{B} = \mathbb{G}$ (com havíem comentat, el mòdul $\mathbb{B}$ no varia). Sigui $\iota' : \mathbb{Z}_n \to \mathbb{G}$ donada per $\iota'(x) = g^x$. Quan h genera $\mathbb{G}$, l'esquema de compromís $c = g^x h^r$ amaga perfectament. Per altra banda, quan h té ordre q, aleshores definim $p' : \mathbb{G} \to \mathbb{Z}_n$ donada per $p'(g^x) = \delta_p x$, que és la projecció de x en el subgrup d'ordre p de $\mathbb{Z}_n$.

SXDH: En aquest cas tenim un grup cíclic d'ordre primer p amb un generador g. Definim $\mathbb{B} = \mathbb{G}^2$, que és un grup abelià amb la multiplicació definida component a component. D'ara endavant definirem $\odot$ com la multiplicació component a component, $\oslash$ la divisió component a component i, si $u = (u^{(1)}, u^{(2)})$ és un element de $\mathbb{G}^2$, aleshores $u^x = ((u^{(1)})^x, (u^{(2)})^x)$. $\mathbb{B}$ és un mòdul sobre $\mathbb{Z}_p$. La clau de compromís contindrà un element $u_1 = (g, h)$ on $h = g^\alpha$ per algun $\alpha \in \mathbb{Z}_p^*$ aleatori. També contindrà un element $u_2 = (u, v)$ que pot ser escollit d'una de les dues maneres: $u_2 = u_1^t$ o bé $u_2 = u_1^t \oslash (1, g)$ per un valor de t escollit aleatòriament en $\mathbb{Z}_p^*$. La primera configuració donarà una clau de lligar mentre que la segona donarà una clau d'amagar. Els dos tipus de claus són indistingibles sota la hipòtesis decisional de Diffie-Hellman.

Per comprometre un element $X \in \mathbb{G}$, primer definim $\iota(X) = (1, X)$. Escollint elements aleatòriament $r_1, r_2 \in \mathbb{Z}_p$, calculem el compromís $c = \iota(X) \odot u_1^{r_1} \odot u_2^{r_1}$. Si $u_2 = u_1^t \oslash (1, g)$ aleshores u_1 i u_2 són linealment independents, per tant són base per $\mathbb{B} = \mathbb{G}^2$, per tant $\iota(G) \subseteq \langle u_1, u_2 \rangle$ i el compromís amaga incondicionalment. Si $u_2 = u_1^t$ aleshores tenim $c = (g^{r_1 + r_2 t}, h^{r_1 + r_2 t} X)$, que és un xifrat ElGamal [Gam85] de X. Definim $p : (c^{(1)}, c^{(2)}) \mapsto c^{(2)}/(c^{(1)})^\alpha$, l'aplicació de desxifrar. Amb aquesta aplicació, el compromís lliga perfectament ja que $p \circ \iota$ és l'aplicació identitat de $\mathbb{G}$ i $p(u_1) = p(u_2) = 1$.

També podem comprometre un exponent $x \in \mathbb{A}' = \mathbb{Z}_p$. Una primera aproximació seria definir la inclusió com $(1, g^x)$ i fer el compromís exactament igual que el compromís d'un element del grup. Ara bé, podem fer-ho d'una altra manera per millorar l'eficiència de l'esquema. Usarem només un element aleatoritzador i definirem la inclusió de manera que continuem tenint les propietats desitjades. Per definir la inclusió, definim $u = u_2 \odot (1, g)$ i $\iota'(x) = u^x$. Per comprometre x usant aleatorietat $r \in \mathbb{Z}_p$ calculem $c = \iota'(x) \odot u_1^r$. És important remarcar que en aquest cas només estem fent servir u_1 com a clau de compromís

pròpiament, mentre que u_2 la fem servir per definir la inclusió ι'. En una clau d'amagar tenim $u = u_1^t$, per tant $u \in \langle u_1 \rangle$ i això implica $\iota'(\mathbb{Z}_p) \subseteq \langle u_1 \rangle$, obtenint un esquema que amaga incondicionalment. Amb una clau de lligar tenim $c = (g^{r+xt}, h^{r+xt} g^x)$, que és un xifrat ElGamal de g^x. Definim $p'(g^{c^{(1)}}, g^{c^{(2)}}) = c^{(2)} - \alpha c^{(1)}$, obtenint que $p' \circ \iota'$ és l'aplicació identitat i $p'(u_1) = 0$, per tant tenim un compromís que lliga perfectament.

DLIN: En un grup on tenim la hipòtesis DLIN, siguin $U = g^\alpha, V = g^\beta$ per uns aleatoris $\alpha, \beta \in \mathbb{Z}_p^*$. Usarem els mòduls $\mathbb{A} = \mathbb{G}$ i $\mathbb{B} = \mathbb{G}^3$, amb la mateixa notació d'abans. La clau de compromís contindrà tres elements $u_1, u_2, u_3 \in \mathbb{B}$. Usem $u_1 = (U, 1, g), u_2 = (1, V, g)$ i u_3 pot ser escollit o bé $u_3 = u_1^r \odot u_2^s$ o bé $u_3 = u_1^r \odot u_2^s \oslash (1, 1, g)$, per obtenir claus de lligar o d'amagar respectivament. La hipòtesis DLIN diu que aquests dos tipus de claus són difícils de distingir.

Per comprometre un element $X \in \mathbb{G}$, procedim de la següent manera. Definim $\iota(X) = (1, 1, X)$. Un compromís es forma escollint $r_1, r_2, r_3 \in \mathbb{Z}_p$ aleatòriament i calculant $c = \iota(X) \odot u_1^{r_1} \odot u_2^{r_2} \odot u_3^{r_3}$. En una clau d'amagar u_1, u_2 i u_3 són linealment independents, per tant formen una base de $\mathbb{B} = \mathbb{G}^3$ i $\iota(G) \subseteq \langle u_1, u_2, u_3 \rangle$, aconseguint un esquema que amaga perfectament. En una clau de lligar tenim $c = (U^{r_1 + r r_3}, V^{r_2 + s r_3}, g^{r_1 + r_2 + (r+s) r_3} X)$, que és un xifrat lineal [BBS04] de X. Definim $p(c^{(1)}, c^{(2)}, c^{(3)}) = c^{(3)} / \left((c^{(1)})^{1/\alpha} \cdot (c^{(2)})^{1/\beta} \right)$. Tenim que $p \circ \iota$ és l'aplicació identitat i $p(u_1) = p(u_2) = p(u_3) = 1$, per tant el compromís lliga perfectament.

A l'hora de comprometre un exponent procedirem de manera similar al cas anterior: usarem u_3 per definir la inclusió i les altres dos claus per aleatoritzar. Si volem comprometre un element $x \in \mathbb{A}' = \mathbb{Z}_p$, definim $u = u_3 \odot (1, 1, g)$ i $\iota'(x) = u^x$. Comprometem x usant aleatorietat r_1, r_2 calculant $c = u^x \odot u_1^{r_1} \odot u_2^{r_2}$. En una clau d'amagar tenim que $u = u_1^r \odot u_2^s$ i per tant $\iota'(\mathbb{Z}_p) \subseteq \langle u_1, u_2 \rangle$ i el compromís amaga perfectament. Amb una clau de lligar, el compromís és $c = (U^{r_1 + rx}, V^{r_2 + sx}, g^{r_1 + r_2 + x(r+s)} g^x)$. Això correspon a un xifrat lineal de g^x. Definim per tant $p'(g^{c^{(1)}}, g^{c^{(2)}}, g^{c^{(3)}}) = c^{(3)} - \frac{1}{\alpha} c^{(1)} - \frac{1}{\beta} c^{(2)}$. Obtenim $p'(u_1) = p'(u_2) = 0$ i $p' \circ \iota'$ és l'aplicació identitat, per tant el compromís lliga incondicionalment.

3.7 Marc general

Ara que ja hem vist com comprometre una variable d'un mòdul $\mathbb{A}_1$ o $\mathbb{A}_2$, falta veure com construir el mòdul $\mathbb{B}_T$ i les aplicacions que falten per tal de poder realitzar les Proves NIWI. Com ja hem introduït abans, per cada equació de les presentades en la secció 3.2 associarem uns mòduls $\mathbb{A}_1, \mathbb{A}_2, \mathbb{A}_T$. Hem vist en la secció anterior que, sigui $\mathbb{A}_1$ (o $\mathbb{A}_2$) un grup $\mathbb{G}_1$, $\mathbb{G}_2$ o bé $\mathbb{Z}_n$, el mòdul $\mathbb{B}_1$ (o $\mathbb{B}_2$) es defineix de la mateixa manera. De fet, donat una hipòtesi concreta, els grups $\mathbb{B}_1, \mathbb{B}_2, \mathbb{B}_T$ queden definits independentment de l'equació de la que volem fer una prova NIWI. De la mateixa manera, l'aplicació bilineal $F : \mathbb{B}_1, \mathbb{B}_2 \to \mathbb{B}_T$ serà independent de l'equació. El que, òbviament, canviarà són les aplicacions $\iota_1, \iota_2, \iota_T$ i, quan s'utilitzin claus de lligar, p_1, p_2, p_T. Com sempre, primer

farem la construcció general i després ja mirarem les particularitzacions (segons hipòtesi i equació).

Igual que al parlar de compromisos es fa la distinció entre claus d'amagar i de lligar, quan parlem del marc general farem la distinció entre l'escenari de consistència i l'escenari de no distinció de testimoni. L'escenari de consistència està relacionat amb les claus de lligar i és el que assegura al verificador que la prova i els compromisos estan unívocament (o quasi) lligats a unes variables que satisfan l'equació original. L'escenari de no distinció de testimoni, que està relacionat amb les claus d'amagar, és el que garanteix que el verificador no podrà distingir quines variables s'han usat per fer els compromisos i les proves, és a dir, quin és el testimoni de verificabilitat de l'equació que es dóna. Anàlogament amb el que passa amb els compromisos, en l'escenari de consistència definirem una aplicació p_T, que no tindrà sentit en l'escenari de no distinció de testimoni, que serà una projecció de $\mathbb{B}_T$ a $\mathbb{A}_T$. En l'escenari de no distinció de testimoni el que haurem de fer és preocupar-nos de que les proves siguin totalment aleatòries, que puguin pertànyer a qualsevol testimoni.

Suposem ara que partim d'uns mòduls $\mathbb{A}_1, \mathbb{A}_2$ i d'unes claus per comprometre elements en $\mathbb{A}_1$ i $\mathbb{A}_2$, que especifiquen $\mathbb{B}_1, \iota_1, p_1, \iota_1', p_1', u_1, \ldots, u_{\hat{m}}$ i $\mathbb{B}_2, \iota_2, p_2, \iota_2', p_2', v_1, \ldots, v_{\hat{n}}$. Els paràmetres comuns han d'incloure també un tercer $\mathcal{R}$-mòdul $\mathbb{B}_T$ amb unes aplicacions $\mathcal{R}$-lineals $\iota_T : \mathbb{A}_T \to \mathbb{B}_T$ i, quan tingui sentit, $p_T : \mathbb{B}_T \to \mathbb{A}_T$. També han d'incloure una aplicació bilineal $F : \mathbb{B}_1, \mathbb{B}_2 \to \mathbb{B}_T$.

El que els demanarem a aquestes aplicacions és que siguin commutatives. Anem a veure el perquè: si prenem una equació, pel que fa a la part de l'esquerra tenim valors i variables en $\mathbb{A}_1, \mathbb{A}_2$ que inclourem o comprometrem i posteriorment el verificador hi farà actuar l'aplicació bilineal F. Pel que fa a la part de la dreta, hi trobem un element resultat de fer actuar f sobre els valors de $\mathbb{A}_1$ i $\mathbb{A}_2$, que el verificador inclourà en $\mathbb{B}_T$. Cal, per tant, que ambdós valors donguin el mateix. És a dir, s'ha de complir que

$$\forall x \in \mathbb{A}_1 \forall y \in \mathbb{A}_2 : F(\iota_1(x), \iota_2(y)) = \iota_T(f(x, y))$$
$$\forall x \in \mathbb{B}_1 \forall y \in \mathbb{B}_2 : f(p_1(x), p_2(y)) = p_T(F(x, y))$$

En la figura 3.2 es pot veure el diagrama de mòduls i aplicacions.

$$\begin{array}{ccccc} \mathbb{A}_1 & \times & \mathbb{A}_2 & \xrightarrow[f]{} & \mathbb{A}_T \\ \iota_1 \downarrow\uparrow p_1 & & \iota_2 \downarrow\uparrow p_2 & & \iota_T \downarrow\uparrow p_T \\ \mathbb{B}_1 & \times & \mathbb{B}_2 & \xrightarrow[F]{} & \mathbb{B}_T \end{array}$$

Figura 3.2: *Diagrama de mòduls i aplicacions*

Per conveniència notacional, si $\vec{x} \in \mathbb{B}_1^n, \vec{y} \in \mathbb{B}_2^n$, definim

$$\vec{x} \bullet \vec{y} = \sum_{i=1}^{n} F(x_i, y_i)$$

que expressat en notació multiplicativa és:

$$\vec{x} \bullet \vec{y} = \prod_{i=1}^{n} F(x_i, y_i)$$

Finalment, els paràmetres també han d'incloure un conjunt de matrius $H_1, \ldots, H_\eta \in \mathrm{Mat}_{\hat{m}\times\hat{n}}(\mathcal{R})$, que totes satisfan $\vec{u}\bullet H_i\vec{v} = 0$ o, en notació multiplicativa, $\vec{u}\bullet exp_M(H_i, \vec{v}) = 1$. El nombre d'aquestes matrius dependrà de la hipòtesis, podent ser aquest nombre zero. De fet, com que només apareixen en la tercera particularització, només les analitzarem en aquest cas. La utilitat d'aquestes matrius la comentarem en la pròxima secció, però bàsicament són matrius que només tenen sentit en l'escenari de no distinció de testimoni i el que fan és aleatoritzar les proves.

3.7.1 Particularitzacions

Per cada particularització, primer es definiran els mòduls, les claus de compromís i les matrius H_i que compleixen el que s'ha dit abans. Posteriorment es definiran les aplicacions bilineals per cada tipus d'equacions. Com que en aquest capítol només volem explicar la construcció de les proves de Groth-Sahai i no la seva correcció (que ja la demostren ells), no reproduirem la demostració de la propietat de commutivitat.

Decisió de subgrup: els paràmetres especifiquen $(p, \mathbb{G}, \mathbb{G}_T, e, g, h)$. Usem $\mathbb{B} = \mathbb{B}_1 = \mathbb{B}_2 = \mathbb{G}$ i $\mathbb{B}_T = \mathbb{G}_T$ i l'aplicació bilineal $F(X, Y) = e(X, Y)$. En l'escenari de no distinció de testimoni fem servir una clau d'amagar h que genera tot $\mathbb{G}$ i, en conseqüència, $e(h, h)$ genera $\mathbb{G}_T$.

Primer veurem l'escenari d'equacions de producte de pairings. En aquest cas tenim $\mathbb{A}_1 = \mathbb{A}_2 = \mathbb{G}$ i $\mathbb{A}_T = \mathbb{G}_T$ i el mapa bilineal $f = e$. Definim l'aplicació $\iota_T : \mathbb{A}_T \to \mathbb{B}_T$ com la identitat, mentre que, si h té ordre q, definim $p_T(z) = z^{\delta_p}$. Notem que en l'escenari de consistència, l'aplicació $p_T \circ \iota_T$ projecta elements al subgrup d'ordre p de $\mathbb{G}_T$.

En l'escenari d'equacions multiescalars, tenim $\mathbb{A}_1 = \mathbb{Z}_n, \mathbb{A}_2 = \mathbb{G}, \mathbb{A}_T = \mathbb{G}$. L'aplicació bilineal és l'exponenciació: $f(x, Y) = Y^x$. Definim $\hat{\iota}_T(Z) = e(g, Z)$ i, quan tingui sentit, $\hat{p}_T(e(g, Z)) = Z^{\delta_p}$. Notem que en l'escenari de consistència, l'aplicació $\hat{p}_T \circ \hat{\iota}_T$ projecta elements al subgrup d'ordre p de $\mathbb{G}$.

Finalment, en el cas d'equacions quadràtiques en $\mathbb{Z}_n$, tenim $\mathbb{A}_1 = \mathbb{A}_2 = \mathbb{A}_T = \mathbb{Z}_n$. L'aplicació bilineal f és la funció multiplicació $f(x, y) = xy \mod n$. Definim $\iota'_T(z) = e(g, g)^z$ i, quan tingui sentit, $p'_T(e(g, g)^z) = \delta_p z$. Notem que en l'escenari de consistència, l'aplicació $p'_T \circ \iota'_T$ projecta elements a un subgrup isomorf a $\mathbb{Z}_p$.

SXDH: els paràmetres especifiquen $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g_1, g_2, u_1, u_2, v_1, v_2)$ on (u_1, u_2) és una clau de compromís pel grup $\mathbb{G}_1$ i (v_1, v_2) és una clau de compromís pel grup $\mathbb{G}_2$, com hem definit en la secció 3.6. Tenim $\mathbb{B}_1 = \mathbb{G}_1^2, \mathbb{B}_2 = \mathbb{G}_2^2$ i definim $\mathbb{B}_T = \mathbb{G}_T^4$ amb la notació esmentada en la secció 3.6. Definim l'aplicació F de la següent manera:

$$F : \mathbb{G}_1^2 \times \mathbb{G}_2^2 \to \mathbb{G}_T^4 \qquad (\begin{pmatrix} X_1 \\ X_2 \end{pmatrix}, \begin{pmatrix} Y_1 \\ Y_2 \end{pmatrix}) \mapsto \begin{pmatrix} e(X_1, Y_1) & e(X_1, Y_2) \\ e(X_2, Y_1) & e(X_2, Y_2) \end{pmatrix}$$

En l'equació de producte de pairings, tenim $\mathbb{A}_1 = \mathbb{G}_1, \mathbb{A}_2 = \mathbb{G}_2, \mathbb{A}_T = \mathbb{G}_T$ i $f(x, y) = e(x, y)$. Les claus de compromís són u_1, u_2 i v_1, v_2 per comprometre elements de $\mathbb{G}_1$ i $\mathbb{G}_2$ respectivament. Pel que fa a les aplicacions ι_T, p_T definim (quan tingui sentit),

$$\iota_T : z \mapsto \begin{pmatrix} 1 & 1 \\ 1 & z \end{pmatrix} \qquad , \qquad p_t(\begin{pmatrix} z_{11} & z_{12} \\ z_{21} & z_{22} \end{pmatrix}) \mapsto z_{22} z_{12}^{-\alpha_1} (z_{21} z_{11}^{-\alpha_1})^{-\alpha_2}.$$

l'aplicació p_T correspon a desxifrar amb ElGamal les columnes de la matriu usant α_1, on $u_1 = (g_1, g_1^{\alpha_1})$ i després desxifrar les files amb ElGamal fent servir α_2, on $v_1 = (g_2, g_2^{\alpha_2})$. Notem que, en l'escenari de consistència, $p_T \circ \iota_T$ és la identitat.

Pel que fa a equacions multiescalars, ens centrarem en el cas $\mathbb{A}_1 = \mathbb{Z}_n, \mathbb{A}_2 = \mathbb{G}_2, \mathbb{A}_T = \mathbb{G}_T$ (el cas on el grup és $\mathbb{G}_1$ es fa de manera idèntica). L'aplicació bilineal és l'exponenciació $f(x, Y) = Y^x$. Usarem u_1 per compromisos d'escalars en $\mathbb{Z}_p$ i v_1, v_2 per compromisos d'elements en $\mathbb{G}_2$. Definim $\hat{\iota}_T(Z) = F(\iota_1'(1), \iota_2(Z)) = F(u, (1, Z))$. Sigui e^{-1} tal que $e^{-1}(e(g_1, Z)) = Z$, aleshores definim $\hat{p}_T = e^{-1}(p_T(z))$. Notem que, en l'escenari de consistència, $\hat{p}_T \circ \hat{\iota}_T$ és la identitat.

Si l'equació és quadràtica en $\mathbb{Z}_p$, aleshores definim les aplicacions $\iota_T'(z) = F(\iota_1'(1), \iota_2'(z) = F(u, v)^z$ i $p_T'(z) = log_{e(g_1, g_2)}(p_t(z))$. Notem que, en l'escenari de consistència, $p_T' \circ \iota_T'$ és la identitat.

DLIN: Els paràmetres especifiquen $(p, \mathbb{G}, \mathbb{G}_T, e, g, u_1, u_2, u_3)$ on (u_1, u_2, u_3) és una clau de compromís per $\mathbb{G}$. Definim $\mathbb{B} = \mathbb{B}_1 = \mathbb{B}_2 = \mathbb{G}^3$ i $\mathbb{B}_T = \mathbb{G}_T^9$, usant la mateixa notació de sempre. En aquest cas usem dues aplicacions bilineals $F, \tilde{F}$. L'aplicació $\tilde{F}$ està definida de la següent manera:

$$\tilde{F} : \mathbb{G}_1^3 \times \mathbb{G}_2^3 \to \mathbb{G}_T^9 \quad (\begin{pmatrix} X_1 \\ X_2 \\ X_3 \end{pmatrix}, \begin{pmatrix} Y_1 \\ Y_2 \\ Y_3 \end{pmatrix}) \mapsto \begin{pmatrix} e(X_1, Y_1) & e(X_1, Y_2) & e(X_1, Y_3) \\ e(X_2, Y_1) & e(X_2, Y_2) & e(X_2, Y_3) \\ e(X_3, Y_1) & e(X_3, Y_2) & e(X_3, Y_3) \end{pmatrix}$$

i la aplicació simètrica F està definida per

$$F(x, y) = \frac{1}{2}\tilde{F}(x, y) + \frac{1}{2}\tilde{F}(y, x)$$

Per les equacions de producte de pairings definim

$$\iota_T(z) = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & z \end{pmatrix}$$

i, quan correspongui,

$$p_T(\begin{pmatrix} z_{11} & z_{12} & z_{13} \\ z_{21} & z_{22} & z_{23} \\ z_{31} & z_{32} & z_{33} \end{pmatrix}) = (z_{33}z_{13}^{-1/\alpha}z_{23}^{-1/\beta})(z_{31}z_{11}^{-1/\alpha}z_{21}^{-1/\beta})^{-1/\alpha}(z_{32}z_{12}^{-1/\alpha}z_{22}^{-1/\beta})^{-1/\beta}.$$

L'aplicació p_T correspon a desxifrar per columnes usant la clau secreta α, β per l'esquema de xifrat lineal [BBS04] i després desxifrar el resultat per files. Notem que, en l'escenari de consistència, $p_T \circ \iota_T$ és la identitat.

Pel que fa a l'aplicació $\tilde{F}$, l'única matriu H tal que $\vec{u}\tilde{\bullet}exp_M(H, \vec{u}) = 1$ és la matriu amb entrades 0 arreu. Aquí, $\tilde{\bullet}$ fa referència al 'producte' $\bullet$ usant l'aplicació $\tilde{F}$. L'aplicació F sí que té solucions no trivials a $\vec{u} \bullet exp_M(H, \vec{u})$, que corresponen a les identitats $F(u_i, u_j) = F(u_j, u_i)$. Les matrius:

$$H_1 = \begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad H_2 = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{pmatrix} \quad H_3 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{pmatrix}$$

formen una bases de totes les matrius tals que $\vec{u} \bullet exp_M(H, \vec{u}) = 1$. Com que aquestes matrius són fixes, no fa falta definir-les explícitament dins els paràmetres.

En el cas d'equacions multiescalars en $\mathbb{G}$, definim $\tilde{\iota}_T(Z) = \tilde{F}(\iota'_1(1), \iota_2(Z)) = \tilde{F}(u, (1, 1, Z))$, $\hat{\iota}_T(Z) = F(\iota'_1(1), \iota_2(Z)) = F(u, (1, 1, Z))$, $\tilde{p}_T(z) = \hat{p}_T(z) = e^{-1}(p_T(z))$, on $e^{-1}(e(g, Z)) = Z$.

En l'escenari de consistència, $\tilde{p} \circ \tilde{\iota}$ i $\hat{p} \circ \hat{\iota}$ són l'aplicació identitat en $\mathbb{G}$. En l'escenari de no distinció de testimonis, l'única solució a $(u_1, u_2)\tilde{\bullet}exp_M(H, \vec{u}) = 1$ és la matriu amb tots els valors zeros mentre que la matriu $H_1 = \begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \end{pmatrix}$ genera totes les matrius H tals que $(u_1, u_2) \bullet exp_M(H, \vec{u}) = 1$.

Finalment, en el cas d'equacions quadràtiques en $\mathbb{Z}_p$, definim les inclusions $\tilde{\iota}'_T(z) = \tilde{F}(\iota'(1), \iota'(z)), \iota'_T(z) = F(\iota'(1), \iota'(z))$ i, quan tingui sentit, $\tilde{p}'_T(z) = p'_T(z) = log_{e(g,g)}(p_T(z))$. Notem que en l'escenari de consistència $p'_T \circ \iota'_T$ i $\tilde{p}'_T \circ \tilde{\iota}'_T$ són l'aplicació identitat en $\mathbb{Z}_p$. Com sempre, per $\tilde{F}$ només tenim matrius trivials H mentre que per F tenim una base no trivial $H_1 = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$

3.8 Demostrant que uns valors compromesos satisfan una equació quadràtica

Recordem primer que les equacions de les quals volem fer-ne Proves NIWI són de la forma

$$\vec{a} \cdot \vec{y} + \vec{x} \cdot \vec{b} + \vec{x} \cdot \Gamma \vec{y} = t,$$

on tenim constants $\vec{a} \in \mathbb{A}_1^n, \vec{b} \in \mathbb{A}_2^m, \Gamma \in \mathrm{Mat}_{m \times n}(\mathcal{R}), t \in \mathbb{A}_T$. Per simplificar aquest apartat, assumirem que només tenim una equació quadràtica. El primer pas és fer compromisos de les variables $\vec{x}$ i $\vec{y}$:

$$\vec{c} = \iota_1(\vec{x}) + R\vec{u} \quad , \quad \vec{d} = \iota_2(\vec{y}) + S\vec{v},$$

amb $R \in \mathrm{Mat}_{m \times \hat{m}}(\mathcal{R}), S \in \mathrm{Mat}_{n \times \hat{n}}(\mathcal{R})$. El provador ha de convèncer al verificador que aquests compromisos contenen $\vec{x} \in \mathbb{A}_1^m, \vec{y} \in \mathbb{A}_2^n$ que satisfan l'equació quadràtica.

Com ja hem dit a la secció 3.3, el que farem és donar uns valors que anomenem Proves NIWI per fer que l'equació principal, en els mòduls $\mathbb{B}_1, \mathbb{B}_2, \mathbb{B}_T$, els compromisos i la prova NIWI, es compleixi.

Per tant primer agafarem l'equació en els mòduls $\mathbb{B}_1, \mathbb{B}_2, \mathbb{B}_T$ i hi inserirem els compromisos, veient quina relació té amb l'equació si hi poséssim les variables sense comprometre. Gràcies a la manera com hem definit totes les aplicacions, trobarem que tota la informació que hem afegit en els compromisos la podem separar. Avaluem doncs

$$\begin{aligned}
&\iota_1(\vec{a}) \bullet \vec{d} + \vec{c} \bullet \iota_2(\vec{b}) + \vec{c} \bullet \Gamma \vec{d} = \\
&= \Big(\iota_1(\vec{a}) \bullet \iota_2(\vec{y}) + \iota_1(\vec{x}) \bullet \iota_2(\vec{b}) + \iota_1(\vec{x}) \bullet \Gamma \iota_2(\vec{y}) \Big) + \\
&+ \iota_1(\vec{a}) \bullet S\vec{v} + R\vec{u} \bullet \iota_2(\vec{b}) + \iota_1(\vec{x}) \bullet \Gamma S\vec{v} + R\vec{u} \bullet \Gamma \iota_2(\vec{y}) + R\vec{u} \bullet \Gamma S\vec{v}
\end{aligned}$$

Usant la propietat commutativa de les aplicacions, la primera part, que està entre parèntesis, equival a $\iota_T(t)$ si es compleix l'equació original. La resta dels termes es poden re-escriure de la següent manera, usant que per tot $\vec{x}, \vec{y}$ es compleix per bilinearitat que $\vec{x} \bullet \Gamma \vec{y} = \Gamma^\top \vec{x} \bullet \vec{y}$:

$$\vec{u} \bullet \Big(R^\top \iota_2(\vec{b}) + R^\top \Gamma \iota_2(\vec{y}) + R^\top \Gamma S \vec{v} \Big) + \Big(S^\top \iota_1(\vec{a}) + S^\top \Gamma^\top \iota_1(\vec{x}) \Big) \bullet \vec{v}.$$

Com que $\vec{u}$ i $\vec{v}$ són públics, és informació que no hem de donar. Una primera aproximació seria definir dos conjunts de valors que composin les proves: $\vec{\pi} = R^\top \iota_2(\vec{b}) + R^\top \Gamma \iota_2(\vec{y}) + R^\top \Gamma S \vec{v}$ i $\vec{\theta} = S^\top \iota_1(\vec{a}) + S^\top \Gamma^\top \iota_1(\vec{x})$.

El verificador haurà de comprovar que es compleix

$$\iota_1(\vec{a}) \bullet \vec{d} + \vec{c} \bullet \iota_2(\vec{b}) + \vec{c} \bullet \Gamma \vec{d} = \iota_T(t) + \vec{u} \bullet \vec{\pi} + \vec{\theta} \bullet \vec{v}$$

A aquesta equació la anomenarem equació de verificació a partir d'ara.

3.8.1 Escenari de no distinció de testimoni

Amb aquesta definició de les proves, tenim assegurada la propietat de correcció, per la manera com ho hem construït, i la propietat de consistència, doncs si apliquem p_1, p_2, p_T a l'equació recuperem l'equació original (o gairebé, en el cas que $p_i \circ \iota_i$ per $i = 1, 2, T$ no sigui la identitat). Ara bé, ens hem d'assegurar que en l'escenari de no distinció de testimoni els valors són tals que qualssevol variables poden estar compromeses.

Primer, cal aleatoritzar els valors de $\vec{\pi}$ i $\vec{\theta}$ sumant i restant els mateixos valors per tal d'evitar que el fet de trencar la prova en dos parts doni informació dels testimonis. Per tant, redefinim $\vec{\pi} = \left(\vec{\pi} - T^\top \vec{v}\right)$ i $\vec{\theta} = \vec{\theta} + T\vec{u}$, on $T \in \mathrm{Mat}_{\hat{n}\times\hat{m}}(\mathcal{R})$. Usant que $u \bullet T^\top \vec{v} = \vec{v} \bullet T\vec{u}$ és evident que aquest canvi no afecta a la correcció de la prova, és a dir, l'equació es continua complint, ni tampoc a la consistència, doncs $p(\vec{u}) = p(\vec{v}) = 0$ en l'escenari de consistència. Amb aquests valors aconseguim que $\vec{\theta}$ estigui uniformement distribuït entre tots els valors possibles.

Ara ens queda observar què li passa a $\vec{\pi}$. Si $\vec{\phi}$ és l'únic valor tal que l'equació de verificació és vàlida donat un valor de $\vec{\theta}$, aleshores està clar que $\vec{\phi}$ no dóna cap informació ja que la unicitat vol dir que qualsevol valor de les variables porta al mateix valor de $\vec{\pi}$ i per tant donat $\vec{\pi}$ no podem distingir quin testimoni s'ha usat. Ara bé, pot ser que donat un valor de $\vec{\theta}$ hi hagi més d'un valor de $\vec{\pi}$ que compleixi l'equació de verificació. En aquest cas, hem de fer que $\vec{\pi}$ tingui distribució uniforme entre totes les possibilitats. Anem a veure com ho fem:

Suposem que dos valors $\vec{\pi}$ i $\vec{\pi}'$ satisfan l'equació de verificació per un $\vec{\theta}$ fixat. Restant una equació de l'altra, obtenim $\vec{u} \bullet (\vec{\pi} - \vec{\pi}') = 0$. Ara bé, havíem dit que en l'escenari de no distinció de testimoni, els vectors $\vec{u}$ generen tot l'espai on viuen $\vec{\pi}$ (efectivament, al estar en una clau d'amagar $\vec{u}$ generen $\mathbb{G}_1^i$, on i és el nombre de vectors que hi ha i, per tant, és igual al nombre de components de $\vec{\pi}$). Com que coneixem les matrius $H_1, \ldots, H_\eta$ que formen una base de totes H tals que $\vec{u} \bullet H\vec{u} = 0$, agafant uns nombres aleatoris $r_1, \ldots, r_\eta$ podem calcular $\vec{\pi}'' = \vec{\pi} + \sum_{i=1}^{\eta} r_i H_i \vec{u}$, que té una distribució uniforme sobre tots els valors de $\vec{\pi}$ possibles que compleixin l'equació de verificació per un valor de $\vec{\theta}$ fixat.

Amb això hem aconseguit que fixat $\vec{\theta}$, $\vec{\pi}$ tingui una distribució uniforme sobre tots els valors possibles. Com que a més fem que $\vec{\theta}$ tingui una distribució uniforme, hem aconseguit que $(\vec{\pi}, \vec{\theta})$ tingui una distribució uniforme sobre tots els valors que compleixen l'equació de verificació.

Finalment, els valors que conformen la prova NIWI són:

$$\vec{\pi}_i = R^\top \iota_2(\vec{b}_i) + R^\top \Gamma_i \iota_2(\vec{y}) + R^\top \Gamma_i S\vec{v} - T_i^\top \vec{v} + \sum_{j=1}^{\eta} r_{ij} H_j \vec{v}$$

$$\vec{\theta}_i = S^\top \iota_1(\vec{a}_i) + S^\top \Gamma_i^\top \iota_1(\vec{x}) + T_i \vec{u}$$

3.8.2 El cas simètric

Un cas interessant és quan $\mathbb{B} = \mathbb{B}_1 = \mathbb{B}_2$, $\hat{m} \geq \hat{n}$ amb $u_1 = v_1, \ldots, u_{\hat{n}} = v_{\hat{n}}$ i per tot $x, y \in \mathbb{B}$ es compleix que $F(x, y) = F(y, x)$. En aquest cas, que anomenem el cas simètric, podem simplificar l'esquema. Primer afegim zeros a $\vec{\theta}$ fins que tingui llargada $\hat{m}$. Anomenem a aquest vector $\vec{\theta'}$ i revelem $\vec{\phi} = \vec{\pi} + \vec{\theta}$.

En la verificació, n'hi ha prou amb comprovar

$$\iota_1(\vec{a}) \bullet \vec{d} + \vec{c} \bullet \iota_2(\vec{b}) + \vec{c} \bullet \Gamma \vec{d} = \iota_T(t) + \vec{u} \bullet \vec{\phi}$$

La construcció de la prova també es pot simplificar ajustant les dimensions de les matrius i eliminant la matriu T.

3.8.3 El cas lineal

Un altre cas interessant és quan l'equació a tractar és lineal, és a dir, tenim $\vec{a} = 0$ i $\Gamma = 0$. En aquest cas, l'equació és simplement

$$\vec{x} \cdot \vec{b} = t$$

L'esquema el podem simplificar escollint $T = 0$, cosa que implica que $\vec{\theta} = 0$ i $\vec{\pi} = R^\top \iota_2(\vec{b}) + \sum_{i=1}^{\eta} r_i H_i \vec{v}$.

3.8.4 Particularitzacions

Abans d'acabar aquesta secció, anem a remarcar un parell de característiques de les particularitzacions.

Pel que fa a la hipòtesi de decisió de subgrup, és clar que l'aplicació F és simètrica i tenim $\mathbb{G}_1 = \mathbb{G}_2$, per tant aplicarem el que s'ha explicat en el subapartat 3.8.2. En aquest cas no hem de fer servir cap matriu H.

Pel que fa a la hipòtesi SXDH ens trobem amb que F no és simètrica. Tampoc hi ha cap matriu aleatoritzadora H, per tant s'usarà la metodologia general.

La particularització sota la hipòtesi DLIN requereix una atenció especial. En aquesta hipòtesi, havíem definit una aplicació F simètrica, que té associades certes matrius H_i aleatoritzadores. Amb aquesta aplicació, la prova NIWI consisteix només d'elements π. Al tractar amb equacions lineals, però, no usarem F sinó que farem servir $\tilde{F}$. Això és degut a que la prova NIWI segueix consistint només d'un vector d'elements ϕ, però al ser una aplicació no simètrica, no apareixen les matrius aleatoritzadores H_i i per tant l'esquema és més eficient.

3.9 Proves de no distinció de testimoni completes

A continuació fem un resum dels passos a seguir per realitzar les proves de no distinció, afegint detalls de les possibles particularitzacions.

Suposarem que els paràmetres CRS, la cadena de referència comuna, estan creats per un dels algorismes K o S, que apareixen en la descripció, i que les sortides dels dos algorismes són indistingibles computacionalment. El primer algorisme dóna uns paràmetres per treballar en l'escenari de consistència mentre que el segon dóna uns paràmetres per treballar en l'escenari de no distinció de testimonis.

Inici(1^λ): l'algorisme d'Inici prové el grup bilineal que es farà servir $g_k = (\mathbb{Z}_n, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e)$ juntament amb una clau secreta sk, que anirà lligada a la hipòtesi sobre la que ens estem basant (per exemple, la factorització de n si és compost).

CRS de consistència - K(g_k, s_k): l'algorisme treu uns paràmetres per treballar en l'escenari de consistència

$$\sigma = (\mathbb{B}_1, \mathbb{B}_2, \mathbb{B}_T, F, \vec{u}, \vec{v}, \{\iota_1, p_1, \iota_2, p_2, \iota_T, p_T, H_1, \ldots, H_\eta\})$$

on s'han de donar totes les aplicacions per incloure elements de qualsevol grup als mòduls corresponents, les claus de compromís, les matrius d'aleatorietat H_i i la descripció de les aplicacions de projecció.

Tot i que els valors de H_i no tenen sentit en aquest escenari, s'usen els de l'altre escenari.

CRS de no distinció de testimonis - S(g_k, s_k): l'algorisme treu uns paràmetres per treballar en l'escenari de no distinció de testimonis

$$\sigma = (\mathbb{B}_1, \mathbb{B}_2, \mathbb{B}_T, F, \vec{u}, \vec{v}, \{\iota_1, p_1, \iota_2, p_2, \iota_T, p_T, H_1, \ldots, H_\eta\})$$

on s'han de donar totes les aplicacions per incloure elements de qualsevol grup als mòduls on toqui, les claus de compromís, les matrius d'aleatorietat H_i i la descripció de les aplicacions de projecció de l'escenari de consistència.

Tot i que les aplicacions p_i no tenen sentit en aquest escenari, s'usa la descripció de l'altre escenari.

Com queda exposat en els dos algorismes, els paràmetres $\mathbb{B}_1, \mathbb{B}_2, \mathbb{B}_T, F$ són fixos. Donat un escenari, els valors de $\vec{u}$ i $\vec{v}$ també són fixos, tot i que el nombre d'elements que s'usin dependrà de l'equació en concret. El que sí que depèn de cada tipus d'equació és les aplicacions d'inclusió i projecció i les matrius H_i.

En els dos escenaris, els valors de H_i només es donen en el cas que siguin no nuls. En les nostres tres hipòtesis, només és per la hipòtesi DLIN. A més, com que són constants tampoc no faria falta incloure'ls.

Evidentment, en el cas que $\mathbb{G}_1 = \mathbb{G}_2 = \mathbb{G}$, el nombre de paràmetres a donar serà molt menor

Prova: L'entrada consisteix dels paràmetres (g_k, σ), una llista d'equacions quadràtiques amb constants $\{(\vec{a}_i, \vec{b}_i, \Gamma_i, t_i)\}_{i=1}^N$ i uns valors de les variables que satisfan l'equació. Notem que els paràmetres $\vec{a}_i, \vec{b}_i, t_i$ poden pertànyer a qualsevol dels grups o bé a $\mathbb{Z}_n$, segons el tipus d'equació.

Primer s'escull aleatòriament $R \in \mathrm{Mat}_{m\times\hat{m}}(\mathcal{R})$ i $S \in \mathrm{Mat}_{n\times\hat{n}}(\mathcal{R})$ per comprometre totes les variables $\vec{c} = \vec{x} + R\vec{u}$ i $\vec{d} = \vec{y} + S\vec{v}$. Cal remarcar que el mateix compromís es manté tota l'estona, encara que la variable aparegui en diferents tipus d'equació.

Per cada equació $(\vec{a}_i, \vec{b}_i, \Gamma_i, t_i)$ es fa una prova com les descrites en la secció 3.8. És a dir, s'agafen $T_i \in \mathrm{Mat}_{\hat{n}\times\hat{m}}(\mathcal{R})$ i $r_{i1}, \ldots, r_{i\eta} \in \mathcal{R}$ i es calcula

$$\vec{\pi}_i = R^\top \iota_2(\vec{b}_i) + R^\top \Gamma_i \iota_2(\vec{y}) + R^\top \Gamma_i S\vec{v} - T_i^\top \vec{v} + \sum_{j=1}^{\eta} r_{ij} H_j \vec{v}$$

$$\vec{\theta}_i = S^\top \iota_1(\vec{a}_i) + S^\top \Gamma_i^\top \iota_1(\vec{x}) + T_i\vec{u}$$

amb les aplicacions que depenen del tipus d'equació.

Verificació Coneguts els paràmetres (g_k, σ), les equacions $\{(\vec{a}_i, \vec{b}_i, \Gamma_i, t_i)\}_{i=1}^N$ i les proves $(\vec{c}, \vec{d}, \{(\vec{pi}_i, \vec{\theta}_i)\}$, comprovar per cada equació:

$$\iota_1(\vec{a}_i) \bullet \vec{d} + \vec{c} \bullet \iota_2(\vec{b}_i) + \vec{c} \bullet \Gamma_i \vec{d} = \iota_T(t_i) + \vec{u} \bullet \vec{\pi}_i + \vec{\theta}_i \bullet \vec{v}$$

amb les aplicacions que depenen del tipus d'equació.

3.10 Proves Sense Coneixement

La diferència entre una prova sense coneixement i una prova NIWI és que la primera ha de ser simulable, és a dir, existeix un simulador que rep certa informació que li permet fer trampes i donar una prova satisfactòria tot i no conèixer cap testimoni que satisfaci l'equació.

El que es fa és modificar el conjunt d'equacions per obtenir-ne un altre en que sí que es pugui trobar un testimoni, només si es coneix la informació que permet fer trampes.

El primer escenari a treballar és el que $\mathbb{A}_1 = \mathcal{R}, \mathbb{A}_2 = \mathbb{A}_T$. L'algorisme S, el de la CRS de no distinció de testimonis, dóna una informació extra τ que permet obrir un compromís a un valor diferent del que s'ha compromès. En particular, τ permet calcular $\vec{s} \in \mathcal{R}^{\hat{m}}$ tal que $\iota_1(1) = \iota_1(0) + \vec{s}^\top \vec{u}$.

Definim $c = \iota_1(1)$, un compromís del valor $\psi = 1$ sense aleatorietat. Re-escrivim l'equació a demostrar:

$$\vec{a}_i \cdot y + f(-\psi, t_i) + \vec{x} \cdot \vec{b}_i + \vec{x} \cdot \Gamma \vec{y} = 0$$

La nova variable ψ permet escollir la solució en la que totes les variables prenen el valor zero. Per simular una solució, comprometem totes les variables com $\vec{c} = \vec{0} + R\vec{u}$, $\vec{d} = \vec{0} + S\vec{v}$ i afegim $c = \iota_1(1) = \iota_1(0) + \sum_{i=1}^{\hat{m}} s_i u_i$ a $\vec{c}$ i construïm proves com les de la secció 3.8. Aquesta prova compleix que per tot i tenim:

$$\iota_1(\vec{a}_i) \bullet \vec{d} + \vec{c} \bullet \iota_2(\vec{b}_i) + F(c, -\iota_2(t_i)) + \vec{c} \bullet \Gamma_i \vec{d} = \iota_T(0) + \vec{u} \bullet \vec{\pi}_i + \vec{\theta}_i \bullet \vec{v}$$

Gràcies a la propietat commutativa, tenim $F(c, \iota_2(t_i)) = F(\iota_1(1), \iota_2(t_i)) = \iota_T(f(1, t_i)) = \iota_T(t_i)$ i per tant compleix l'equació que el verificador mirarà.

En resum, el simulador ha fet una prova de tal manera que els valors $\vec{c}, \vec{d}, \vec{\pi}, \vec{\theta}$ satisfan l'equació de verificació gràcies a que el compromís del zero que havia realitzat és equivalent a la inclusió de l'ú.

L'explicat abans no cobreix el cas d'equacions de producte de pairings. Això és perquè en general, no podem tractar l'element de $\mathbb{A}_T = \mathbb{G}_T$ com si fos un element de $\mathbb{A}_2 = \mathbb{G}_2$ i per tant no podem pretendre que ve d'una aplicació bilineal. Ara bé, si $t_T = 1$, aleshores l'equació de producte de pairings té la solució trivial en que totes les variables són 1. Si $t_T = \prod e(g_i, h_i)$, aleshores podem definir variables Z_i i modificar les equacions: les equacions producte de pairings seran del tipus $\prod e(g_i, Y_i) \cdot \prod e(X_i, h_i) \cdot \prod \prod e(X_i, Y_j)^{\gamma_{ij}} \prod e(Z_i, h_i) = 1$ i afegirem equacions multi-escalars $Z_i = h_i$. Amb aquestes noves equacions estem en el cas on sí que sabem simular una prova vàlida.

3.11 Repassant les proves de Groth-Sahai

En el PKC 2010, Ghadafi et al. van publicar un article amb el títol *Groth-Sahai proofs revisited* [GSW10], que fa unes observacions força interessants sobre l'article de Groth i Sahai.

En primer lloc, corregeix unes errades en la definició de certes aplicacions que apareixia en l'article original de Groth i Sahai. En l'explicació feta fins ara ja s'han tingut en compte aquestes correccions.

En l'article revisa els tipus de pairings i grups bilineals definits en [GPS08]. En distingeix tres tipus:

- **Tipus-1:** Aquest és el cas simètric, on $\mathbb{G}_1 = \mathbb{G}_2$
- **Tipus-2:** En aquest cas tenim $\mathbb{G}_1 \neq \mathbb{G}_2$ però existeix un isomorfisme eficientment computable $\psi : \mathbb{G}_2 \to \mathbb{G}_1$ on $\psi(g_2) = g_1$ on g_1 i g_2 són generadors de $\mathbb{G}_1$ i $\mathbb{G}_2$ respectivament.

- **Tipus-3:** En aquest cas $\mathbb{G}_1 \neq \mathbb{G}_2$ i no es coneix cap isomorfisme eficientment computable.

D'aquesta classificació en deduïm que la hipòtesi SXDH només és aplicable a grups del tercer tipus, doncs en els altres el problema Diffie-Hellman decisional és fàcil com a mínim en un dels dos grups.

En l'article s'explica que tant els grups bilineals d'ordre compost (en el que es basa la primera particularització) com els grups bilineals del primer tipus tenen poc impacte pràctic, doncs són ineficients.

Ghadafi et al. defineixen una extensió asimètrica de la hipòtesi DLIN, on es treballa en grups bilineals del segon o tercer tipus i s'assumeix que el problema DLIN és difícil tant en $\mathbb{G}_1$ com en $\mathbb{G}_2$.

Finalment, fan un esbós de com treballar en grups del segon tipus suposant que el problema Diffie-Hellman decisional és difícil en $\mathbb{G}_1$ i el DLIN és difícil en $\mathbb{G}_2$.

Capítol 4

ABS en el model estàndard i estructura d'accés genèrica

4.1 Introducció

La firma basada en atributs (d'ara endavant ABS, de l'anglès Attribute-Based Signature) és una primitiva criptogràfica recent que parteix de les firmes digitals. Els esquemes de firma ABS són una extensió natural dels esquemes de firma basats en identitats, en els quals la clau pública del firmant és la seva identitat. En un esquema ABS la clau pública del firmant és un conjunt d'atributs, enlloc d'una identitat. El firmant ha obtingut una clau secreta per part d'una entitat, que acredita que el firmant té aquests atributs que pretén tenir.

En una firma ABS el verificador comprova la validesa de la firma i, si és el cas, queda convençut que la firma ha estat generada per una persona que té certs atributs. Donat que la firma no dóna cap coneixement sobre la identitat del firmant, aquest esquema garanteix cert anonimat. A més a més, un esquema ABS pot estar construït de manera que hi hagi famílies d'atributs vàlides i el verificador queda convençut que el firmant té atributs d'alguna d'aquestes famílies sense saber quina.

Hi ha molts escenaris on aquests esquemes es poden aplicar. Per exemple, un escenari pot ser la publicació d'una nota de premsa tècnica en el que es vulgui preservar l'anonimat de l'autor. Per demostrar que l'autor té coneixements suficients per firmar el missatge, el pot firmar assegurant que o bé és professor de la universitat i és del departament de matemàtiques o bé és el cap d'una empresa i té un doctorat en matemàtica aplicada. Al verificar la firma, l'única informació que se'n desprén és que el firmant encaixa en algun dels dos perfils.

Un altre escenari d'aplicació és en l'autenticació basada en privilegis. En una firma digital bàsica hauríem de comprovar a qui correspon la clau de verificació i comprovar que la persona té els privilegis suficients. Utilitzant un esquema ABS n'hi ha prou amb enviar

un repte a ser firmat i comprovar que la firma està realitzada amb atributs corresponents a certs privilegis.

És important observar que les nocions de seguretat quan es consideren firmes ABS són molt més complexes que en les firmes estàndard. Amb les firmes de grup o firmes en anell apareix la noció d'anonimat i de no enllaçament. Els esquemes ABS hereten aquestes propietats i, a més, fan aparèixer el concepte de resistència a col·lisió ja que els atributs han d'estar relacionats amb cada persona.

4.1.1 La nostra contribució

En aquest article desenvolupem un ABS a partir de l'esquema de firma en anell de Shacham-Waters [SW07]. Aconseguim l'esquema fent dos passos (implícitament): primer construïm una firma distribuïda i posteriorment aconseguim la resistència a col·lisions.

La demostració de la seguretat de l'esquema està basada en el model estàndard i permet diferents esquemes d'estructures d'accés per definir famílies d'atributs autoritzades. Per una banda, es poden construir esquemes de llindar, cosa que ja s'aconsegueix en [LAS+10, SSN09] (si bé hi ha altres resultats sobre esquemes ABS, la seguretat d'aquests no es basa en el model estàndard i per tant no els tenim en compte en el moment de fer comparacions). A més a més, el nostre esquema permet realitzar un cas particular d'estructures multipartites. Finalment, a costa d'augmentar el tamany de la firma, es poden fer estructures d'accés d'espai vectorial.

Una altra característica del nostre esquema es la possibilitat de rastrejar als firmants en cas que sigui necessari. Aquesta característica no és possible en els treballs anteriors [LAS+10, SSN09] donada la seva construcció.

Finalment, la seguretat obtinguda en el nostre esquema és lleugerament superior a la de [LAS+10, SSN09]. En el seu esquema el nivell de seguretat és no falsificació existencial amb política d'accés selectiva i atac de missatge escollit adaptativament. Això vol dir que el conjunt d'atributs autoritzats per realitzar la falsificació està fixat des del principi del joc de seguretat. En el nostre cas permetem que el conjunt d'atributs autoritzats en la falsificació no estigui fixat, cosa que augmenta la seguretat de l'esquema.

4.1.2 Treballs previs

Firmes basades en identitats

Les firmes basades en identitats és una primitiva criptogràfica definida per Shamir [Sha84] en la que la clau pública d'un usuari és el hash d'una cadena de caràcters que representa la seva identitat. En aquests esquemes no fa falta comprovar el certificat de la clau pública de l'usuari ja que la pròpia clau pública actua de certificat: l'usuari té la clau secreta corresponent només si aquella és la seva identitat.

Firma basada en identitats borrosa

Les firmes basades en identitats borroses (en anglès, Fuzzy Identity-based Signature) [CZCG09] permeten generar firmes amb alguns dels atributs del firmant. Aquests esquemes no consideren cap noció de privacitat, a diferència dels esquemes ABS. Recordem que en els esquemes ABS una firma ha de convèncer al verificador que el firmant té uns atributs que satisfan certa estructura, i res més. Les firmes basades en identitats borroses, que les podem considerar com firmes ABS sense privacitat, poden ser construïdes trivialment gràcies a la metodologia proposada per Galindo et al. [GHK06].

Firma d'anell i firma de grup

Aquestes dues primitives permeten a un usuari realitzar firmes en nom d'un conjunt d'usuaris sense revelar la identitat del firmant. La diferència entre la firma d'anell [RST01] i la firma de grup [CvH91] està en la formació del grup: en una firma d'anell el conjunt es forma en el moment de firmar el missatge mentre que en la firma del grup el conjunt d'usuaris està format abans de fer la firma. Una característica remarcable de les firmes d'anell és que per realitzar una firma d'anell és necessari conèixer les claus públiques dels usuaris dins del conjunt.

Firma de malla

Les firmes de malla (en anglès, Mesh Signature) [Boy07] són una extensió de les firmes d'anell on cada usuari té certs atributs. Les estructures d'accés possibles poden ser, en conseqüència, molt més complexes que en les firmes d'anell. La gran diferència entre una firma de malla i una firma ABS és que en la primera no hi ha resistència a col·lisió, per tant els usuaris poden usar les seves claus secretes per crear firmes que cap d'ells podria crear per separat.

Xifrat basat en atributs

És interessant fer una menció al xifrat basat en atributs, anàleg a la firma basada en atributs. El xifrat basat en atributs és introduït per Sahai i Waters [SW05] com una extensió del xifrat basat en identitats. A diferència dels esquemes de firma, en el xifrat basat en atributs no apareix cap noció d'anonimat. Per altra banda, la noció de seguretat és força diferent: un adversari té èxit si aconsegueix distingir el missatge que s'ha xifrat. Una altra diferència entre xifrat i firma és que actualment existeixen xifrats basats en atributs [HLR10] en el qual el missatge xifrat té mida constant respecte el nombre d'atributs, mentre que en tots els esquemes de firma basats en atributs existents el tamany de la firma depèn linealment del nombre d'atributs.

4.2 Preliminars

4.2.1 Sobre els compromisos

Al llarg d'aquest capítol usarem fortament els compromisos i les proves de Groth-Sahai, majoritàriament en la particularització de decisió de subgrup. D'ara endavant, denotarem $Com(x)$ per un compromís del valor de x, i π_{exp} per una prova NIWI referent a l'equació que involucra exp. Serà útil recordar que, en l'escenari de no distinció de grup, el compromís de $x \in \mathbb{G}$ es construeix com $Com(x) = xh^r$ per un $r \in \mathbb{Z}_n$.

A banda de les proves explicades en el capítol anterior, usarem unes proves amb no distinció de testimoni específiques per comprometre un 0 o un 1 a l'exponent d'una base coneguda. Aquestes proves són proposades per Groth et al. en l'article [GOS06]. Aquestes proves, basades en la hipòtesi de la decisió de subgrup, són molt semblants a les proves de Groth-Sahai pel que fa a l'escenari i les propietats. L'únic que les diferencien de les proves del capítol anterior és la manera de construir-les: el compromís es calcula com $C = u^b h^r$, la prova és $\pi = (u^{2b-1}h^r)^r$ on $b \in \{0, 1\}$ i l'equació de verificació és $e(C, C/u) \stackrel{?}{=} e(h, \pi)$. La demostració de que aquesta prova té les propietats per ser una prova amb no distinció de testimoni, s'expliquen en l'article [GOS06]. Amb una prova d'aquest tipus, el verificador queda convençut que el compromís conté algun dels dos valors 0 o 1 a l'exponent de la base u.

4.2.2 Hipòtesi Diffie-Hellman computacional

Una hipòtesi en la qual basem la seguretat de l'esquema és la hipòtesi Diffie-Hellman computacional (en anglès, Computational Diffie-Hellman).

Suposem que tenim un grup bilinear $(n, \mathbb{G}, \mathbb{G}_T, e, g)$. Sigui $\mathbb{G}_p$ el subgrup d'ordre p de $\mathbb{G}$ amb un generador $g_p \in \mathbb{G}_p$. La hipòtesi Diffie-Hellman computacional en $\mathbb{G}_p$ diu que no existeix cap algorisme $\mathcal{A}$ que s'executi en temps polinòmic (respecte un paràmetre de seguretat) i que donada una tupla (g_p, g_p^a, g_p^b), la descripció del grup bilinear i la factorització (p, q) de l'ordre n pugui calcular g_p^{ab} amb probabilitat no negligible (respecte el paràmetre de seguretat). La probabilitat és sobre el generador $g_p \in \mathbb{G}_p$, els valors a, b i la aleatorietat de $\mathcal{A}$.

4.3 Definicions

Un esquema ABS és un esquema de firma que expressa que la firma ha estat generada per un usuari amb certs atributs però no revela quin usuari ha generat la firma ni quins atributs s'han fet servir. Un esquema ABS ha de satisfer tres propietats de seguretat: anonimat, no enllaçament i no falsificació. La no falsificació es satisfà si cap adversari no pot construir una firma vàlida sota algun conjunt d'atributs que no posseeix. Per altra

banda l'anonimat es satisfà si cap adversari no pot distingir qui ha firmat el missatge o quins atributs s'han utilitzat. Pel que fa a l'anonimat agafarem un model prou fort, on l'adversari pot corrompre qualsevol signant (inclús el mateix que ha produït la firma). El no enllaçament es satisfà si cap adversari pot relacionar dues firmes fetes per la mateixa persona.

A banda d'aquestes propietats de seguretat, un esquema ABS també pot ser rastrejable. Això vol dir que hi pot haver una entitat que pugui trencar l'anonimat del signant quan sigui necessari. En aquest cas, es defineix una propietat de seguretat del rastreig, que garanteix que cap adversari pot fer una firma que rastregi cap a una altra persona que no sigui ell.

4.3.1 Definició de l'esquema

Un esquema ABS consisteix de cinc algorismes: l'**Inici**, la generació de claus (**GenClau**) la **Firma**, la **Verificació** i el **Rastreig**. Formalment es defineix com:

- **Inici**(1^λ). L'algorisme d'inicialització pren com a entrada un paràmetre de seguretat i dóna com a sortida uns paràmetres públics PP, una clau secreta mestra MK i una clau de rastreig TK (de l'anglès, Tracing Key).
- **GenClau**(A, MK, PP). L'algorisme de generació de claus pren com a entrada un conjunt d'atributs A, la clau secreta mestra MK i els paràmetres públics PP i dóna com a sortida una clau secreta SK_A. La clau secreta conté un valor que identifica únicament la clau amb el firmant que demana la clau.
- **Firma**(M, Γ, SK_A, PP). L'algorisme de firma pren com a entrada un missatge M, una estructura d'accés Γ, una clau secreta SK_A i els paràmetres públics PP i dóna com a sortida una firma σ. El valor que identifica la clau secreta amb l'usuari s'usa en la firma.
- **Verificació**(σ, M, Γ, PP). L'algorisme de verificació pren com a entrada una firma σ sobre un missatge M, una estructura d'accés Γ i els paràmetres públics PP i accepta o rebutja la firma, depenent de la seva validesa.
- **Rastreig**(σ, TK, PP). L'algorisme de rastreig pren com a entrada una firma σ, una clau de rastreig TK i els paràmetres públics PP, i dóna com a sortida el valor de la clau secreta que permet identificar al firmant.

4.3.2 Definició de la seguretat

Quan considerem la **no falsificació** d'un esquema de firma, podem considerar diferents nivells de seguretat. En aquest cas definim el nivell més fort de seguretat, que és no falsificació existencial i missatges escollits adaptativament. En altres treballs el conjunt

d'atributs autoritzats de la falsificació es coneix abans de començar el joc, cosa que en el nostre esquema no passa. Per això diem que la seguretat és més gran que la obtinguda per les altres propostes. La falsificació està definida mitjançant el següent joc entre un reptador $\mathcal{C}$ i un adversari $\mathcal{A}$:

Inici: $\mathcal{C}$ usa l'algorisme **Inici** de la firma, es guarda la clau secreta mestra MK i li dona els paràmetres públics PP a $\mathcal{A}$. La clau de rastreig TK no es guarda enlloc.

Peticions: Adaptativament, $\mathcal{A}$ pot fer qualssevol peticions de les següents:

- Petició de clau privada: $\mathcal{A}$ pot fer una petició per una clau privada amb un conjunt d'atributs A.
- Petició de firma: $\mathcal{A}$ pot demanar una firma per un missatge M i una estructura d'accés Γ.

$\mathcal{C}$ respon a cada petició abans d'acceptar la següent. $\mathcal{A}$ fa un total de q_E peticions de clau privada i q_S peticions de firma.

Sortida: Finalment, $\mathcal{A}$ envia a $\mathcal{C}$ una tupla $(\sigma^*, M^*, \Gamma^*)$ i guanya el joc si (1) $\mathcal{A}$ no ha fet cap petició per una clau privada tal que el conjunt d'atributs A de la petició estigui autoritzat per Γ^*, i (2) $\mathcal{A}_1$ no ha demanat una firma pel missatge M^* i l'estructura d'accés i (3) l'equació de verificació es compleix.

Sigui èxit l'event en que $\mathcal{A}$ guanya el joc de falsificació. L'avantatge de $\mathcal{A}$ es defineix com $\mathrm{Adv}_{\mathcal{A}}^{\text{ABS-EUF}} = Pr(\text{èxit})$, on la probabilitat es pren sobre el llançament de monedes que fan $\mathcal{A}$ i $\mathcal{C}$.

Definició 1. *Un adversari $\mathcal{A}$ trenca l'esquema ABS amb paràmetres (t, ϵ, q_E, q_S) si $\mathcal{A}$ corre amb temps com a molt t, fa com a molt q_E peticions de clau privada i q_S peticions de firma, i $Adv_{\mathcal{A}}^{ABS\text{-}EUF}$ és com a mínim ϵ. Un esquema ABS és no falsificable amb paràmetres (t, ϵ, q_E, q_S) si no existeix cap adversari que trenqui l'esquema amb paràmetres (t, ϵ, q_E, q_S).*

La **resistència a col·lisió** és un requisit molt important en els criptosistemes basats en atributs, en els quals un grup d'usuaris poden compartir les seves claus secretes i intentar firmar un missatge amb una estructura d'accés per la qual cap dels usuaris està autoritzat. És important remarcar que la pròpia definició de no falsificació garanteix la resistència a col·lisió.

La propietat de **rastreig** és la que garanteix que qualsevol firma pot ser rastrejada fins al firmant i que no es pot produir una firma que rastreji cap a una altra persona. La seguretat es defineix amb el següent joc:

Inici: l'inici és el mateix que el del joc de falsificació, pero ara el reptador es guarda la clau de rastreig.

Peticions: les peticions es tracten de la mateixa manera que en el joc de falsificació, amb l'afegit que es poden fer q_T peticions de rastreig.

Sortida: Finalment, $\mathcal{A}$ envia a $\mathcal{C}$ una tupla $(\sigma^*, M^*, \Gamma^*)$ i guanya el joc si (1) $\mathcal{A}$ aconsegueix fer una firma que no es rastregi cap a ell (és a dir, cap a cap valor relacionable amb la seva identitat), i (2) l'equació de verificació es compleix.

Sigui èxit l'event en que $\mathcal{A}$ guanya el joc de rastreig. L'avantatge de $\mathcal{A}$ es defineix com $\text{Adv}_{\mathcal{A}}^{\text{ABS-TRA}} = Pr(\text{èxit})$, on la probabilitat es pren sobre el llançament de monedes que fan $\mathcal{A}$ i $\mathcal{C}$.

Definició 2. *Un adversari $\mathcal{A}$ trenca l'esquema ABS amb paràmetres $(t, \epsilon, q_E, q_S, q_T)$ si $\mathcal{A}$ corre amb temps com a molt t, fa com a molt q_E peticions de clau privada i q_S peticions de firma, q_T peticions de rastreig i $Adv_{\mathcal{A}}^{ABS\text{-}TRA}$ és com a mínim ϵ. Un esquema ABS és rastrejable amb paràmetres $(t, \epsilon, q_E, q_S, q_T)$ si no existeix cap adversari que trenqui l'esquema amb paràmetres $(t, \epsilon, q_E, q_S, q_T)$.*

L'**anonimat** es defineix mitjançant el següent joc entre un reptador $\mathcal{C}$ i un adversari $\mathcal{A}$:

Inici: $\mathcal{C}$ usa l'algorisme **Inici** de la firma, es guarda la clau secreta mestra MK i li dona els paràmetres públics PP a $\mathcal{A}$.

Peticions: Adaptativament, $\mathcal{A}$ pot fer qualssevol peticions de les següents:

- Petició de clau privada: $\mathcal{A}$ pot fer una petició per una clau privada amb un conjunt d'atributs A.
- Petició de firma: $\mathcal{A}$ pot demanar una firma per un missatge M i una estructura d'accés Γ.
- Petició de rastreig d'una firma: $\mathcal{A}$ pot demanar que donada una firma se li retorni el valor relacionat amb la identitat usada en la firma.

$\mathcal{C}$ respon a cada petició abans d'acceptar la següent. $\mathcal{A}$ fa un total de q_E peticions de clau privada i q_S peticions de firma, i q_T peticions de rastreig.

Repte: $\mathcal{A}$ publica el repte $(M, \Gamma, A_{i_0}, A_{i_1})$ on A_{i_0} i A_{i_1} són conjunts d'atributs i Γ és una estructura d'accés tal que $A_{i_0}, A_{i_1} \in \Gamma$. El reptador escull una moneda aleatòria $b \in \{0, 1\}$, calcula $\sigma = \mathsf{Firma}(M, \Gamma, SK_{i_b}, \text{PP})$ i entrega σ a $\mathcal{A}$.

Sortida: Finalment, $\mathcal{A}$ dóna com a sortida b' i guanya el joc si $b = b'$.

Sigui èxit l'event en que $\mathcal{A}$ guanya el joc d'anonimat. L'avantatge de $\mathcal{A}$ es defineix com $\text{Adv}_{\mathcal{A}}^{\text{ABS-AN}} = |Pr(\text{èxit}) - 1/2|$, on la probabilitat es pren sobre el llançament de monedes que fan $\mathcal{A}$ i $\mathcal{C}$.

Definició 3. *Un adversari $\mathcal{A}$ trenca l'esquema ABS amb paràmetres (t, ϵ, q_E, q_S) si $\mathcal{A}$ corre amb temps com a molt t, fa com a molt q_E peticions de clau privada i q_S peticions*

de firma, q_T peticions de rastreig i $Adv_{\mathcal{A}}^{ABS\text{-}AN}$ és com a mínim ϵ. Un esquema ABS és anònim amb paràmetres $(t, \epsilon, q_E, q_S, q_T)$ si no existeix cap adversari que trenqui l'esquema amb paràmetres $(t, \epsilon, q_E, q_S, q_T)$.

Una altra propietat que compleix el nostre esquema és la propietat de **no enllaçament**. Un esquema de firma és no enllaçable quan donats dos missatges no es pot dir si estan firmats per la mateixa persona o no. Aquesta propietat es defineix pel següent joc de seguretat:

Inici: $\mathcal{C}$ usa l'algorisme **Inici** de la firma, es guarda la clau secreta mestra MK i li dona els paràmetres públics PP a $\mathcal{A}$.

Peticions: Adaptativament, $\mathcal{A}$ pot fer qualssevol peticions de les següents:

- Petició de clau privada: $\mathcal{A}$ pot fer una petició per una clau privada amb un conjunt d'atributs A.
- Petició de firma: $\mathcal{A}$ pot demanar una firma per un missatge M i una estructura d'accés Γ.
- Petició de rastreig d'una firma: $\mathcal{A}$ pot demanar que donada una firma se li retorni el valor relacionat amb la identitat usada en la firma.

$\mathcal{C}$ respon a cada petició abans d'acceptar la següent. $\mathcal{A}$ fa un total de q_E peticions de clau privada i q_S peticions de firma, i q_T peticions de rastreig.

Repte: $\mathcal{A}$ publica el repte $(M_1, \sigma_1, M_2, \Gamma, A)$ on A és un conjunts d'atributs, Γ és una estructura d'accés tal que $A \in \Gamma$ i σ_1 és la firma del missatge M_1. El reptador escull una moneda aleatòria $b \in \{0, 1\}$ i si surt $b = 0$ firma el missatge M_2 amb la mateixa identitat (el mateix valor relacionat amb la identitat) que σ_1; altrament firma el missatge M_2 amb un altre valor, relacionat amb qualsevol altra identitat.

Sortida: Finalment, $\mathcal{A}$ dóna com a sortida b' i guanya el joc si $b = b'$.

Sigui $\grave{\text{e}}\text{xit}$ l'event en que $\mathcal{A}$ guanya el joc d'enllaçament. L'avantatge de $\mathcal{A}$ es defineix com $\text{Adv}_{\mathcal{A}}^{\text{ABS-NE}} = |Pr(\grave{\text{e}}\text{xit}) - 1/2|$, on la probabilitat es pren sobre el llançament de monedes que fan $\mathcal{A}$ i $\mathcal{C}$.

Definició 4. *Un adversari $\mathcal{A}$ trenca l'esquema ABS amb paràmetres (t, ϵ, q_E, q_S) si $\mathcal{A}$ corre amb temps com a molt t, fa com a molt q_E peticions de clau privada i q_S peticions de firma, q_T peticions de rastreig i $Adv_{\mathcal{A}}^{ABS\text{-}NE}$ és com a mínim ϵ. Un esquema ABS és no enllaçable amb paràmetres $(t, \epsilon, q_E, q_S, q_T)$ si no existeix cap adversari que trenqui l'esquema amb paràmetres $(t, \epsilon, q_E, q_S, q_T)$.*

4.4 Proposta d'esquema ABS en el model de l'oracle aleatori, amb no enllaçament i en el cas llindar

A continuació es descriuen els diferents algorismes que intervenen en la nostra proposta d'esquema ABS. Podem pensar que cada algorisme l'executa una entitat diferent. Parlarem, per exemple, de l'entitat rastrejadora per referir-nos a l'algorisme de rastreig.

Inici(λ): L'algorisme d'inicialització primer genera un grup bilinear $\mathbb{G}$ d'ordre compost $n = pq$ on p i q són primers de tamany $\Theta(\lambda)$. A més, escull elements aleatoris $g, w \in \mathbb{G}, h \in \mathbb{G}_q$, on $\mathbb{G}_q$ és el subgrup d'ordre q de $\mathbb{G}$, i $s \in \mathbb{Z}_n$ i dues funcions hash $H_1, H_2 : \{0,1\}^* \to \mathbb{G}$. Finalment escull un esquema de firma automòrfic[1] (basat en el mateix grup bilinear) amb clau pública PK_σ i clau secreta SK_σ. Publica els paràmetres públics PP i es guarda la clau secreta mestra MK:

$$PP = (n, \mathbb{G}, \mathbb{G}_T, e, g, g_1 = g^s, h, h_1 = h^s, w, H_1, H_2, PK_\sigma), MK = (s, SK_\sigma)$$

També se li dona la clau secreta de rastreig, $TK = p, q$, a l'entitat rastrejadora.

GenClau(A, MK, PP): L'algorisme de generació de claus pren com a entrada la identitat de l'usuari, el conjunt d'atributs A de l'usuari i la clau secreta màster MK.

L'entitat proveïdora d'atributs escull un element aleatori $K \in \mathbb{G}$ i el firma, obtenint $\sigma_K \in \mathbb{G}$. Per cada atribut $\mathsf{at}_i \in A$ que tingui l'usuari, l'autoritat escull un element aleatori $r_i \in \mathbb{Z}_n$ i defineix la clau privada de l'atribut $SK_i = (E_i, G_i) = (H_1(\mathsf{at}_i)^s K^{r_i}, g^{r_i})$. L'usuari rep $SK_A = (K, \sigma_K, \{SK_i\}_{\mathsf{at}_i \in A})$.

Firma(M, l, B, SK_A, PP) L'algorisme de firma pren com a entrada un missatge M, un llindar l, el conjunt B d'atributs vàlids per firmar i la claus privada de l'usuari SK_A. L'usuari selecciona un conjunt autoritzat d'atributs A_S, és a dir, un subconjunt de $A \cap B$ de cardinal l. Per una banda, per tots els atributs de B, escull un nombre aleatori $z_i \in \mathbb{Z}_n$ i calcula

$$C_i = (H_1(\mathsf{at}_i)/w)^{f_i} h^{z_i} \text{ i } \pi_i = ((H_1(\mathsf{at}_i)/w)^{2f_i - 1} h^{z_i})^{z_i}$$

on $f_i = 1$ si $\mathsf{at}_i \in A_S$ i $f_i = 0$ altrament.

Per altra banda calcula $H_m = H_2(M, l, B)$, escull un nombre aleatori $t \in \mathbb{Z}_n$ i calcula

$$\sigma_1 = \left(\prod_{\mathsf{at}_i \in A_S} E_i \right) H_m^t h_1^z \text{ , } \sigma_2 = g^t \text{ i } \sigma_3 = \prod_{\mathsf{at}_i \in A} G_i.$$

on $z = \sum_{\mathsf{at}_i \in B} z_i$. Finalment, la firma és:

$$\sigma = (\sigma_1, \sigma_2, \sigma_3, \{(C_i, \pi_i)\}_{\mathsf{at}_i \in B}, K, \sigma_K)$$

[1] Un esquema de firma automòrfic és aquell en el qual la clau de verificació, el missatge i la firma són elements d'un grup bilinear i la verificació consisteix en avaluar una sèrie d'equacions de producte de pairings. En la secció 4.7.1 hi ha un exemple d'un esquema de firma automòrfic.

Verificació (σ, M, l, B, PP): L'algorisme de verificació pren com a entrada una firma σ, el missatge M, el llindar l i el conjunt d'atributs B. Procedeix de la següent manera:

1. Per tot $\mathsf{at}_i \in B$ comprova si es compleix $e(C_i, C_i/(H_1(\mathsf{at}_i)/w)) \stackrel{?}{=} e(h, \pi_i)$.
2. Comprova si es compleix $e(\sigma_1, g) \stackrel{?}{=} e(w^l \prod_{\mathsf{at}_i \in B} C_i, g_1) e(H_m, \sigma_2) e(K, \sigma_3)$
3. Comprova que σ_K és la firma de K.
4. Si tots els passos són correctes, la firma és vàlida, altrament no ho és.

Teorema 1. *L'esquema té la propietat de correcció.*

Prova. Surt directament de desenvolupar la part esquerra de l'equació:

$$\begin{aligned} e(\sigma_1, g) &= e(\left(\prod_{\mathsf{at}_i \in A} E_i\right) H_m^t h_1^z, g) \\ &= e(\prod_{\mathsf{at}_i \in A} \mathsf{at}_i h^z, g^s) e(H_m, g^t) e(K, g^r) \\ &= e(w^l \prod_{\mathsf{at}_i \in B} C_i, g_1) e(H_m, \sigma_2) e(K, \sigma_3) \end{aligned}$$

4.4.1 Aconseguint no enllaçament

És clar que els elements σ_3, K i σ_K possibiliten enllaçar dos firmes de missatges diferents fetes per un mateix usuari. El que fem és utilitzar les proves de Groth-Sahai per comprometre aquests dos valors.

Anomenem $Com(\sigma_3)$, $Com(K)$ i $Com(\sigma_K)$ als compromisos d'aquests valors, π_σ a la prova NIWI de la segona equació de la verificació i π_K la prova NIWI de la firma de K. D'aquesta manera, la firma serà

$$\sigma = (\sigma_1, \sigma_2, Com(\sigma_3), \{(C_i, \pi_i)\}_{\mathsf{at}_i \in B}, Com(K), Com(\sigma_K), \pi_K, \pi_\sigma)$$

la segona equació de verificació queda com

$$e(\sigma_1, g) \stackrel{?}{=} e(w^l \prod_{\mathsf{at}_i \in B} C_i, g_1) e(H_m, \sigma_2) e(Com(K), Com(\sigma_3)) e(h, \pi_\sigma)$$

i també s'ha de verificar la firma sobre K, usant la verificació d'una prova de Groth-Sahai. Les equacions a verificar dependran de la firma automòrfica escollida, però pel fet

d'escollir una firma automòrfica sabrem que les equacions de verificació seran productes de pairings i podrem aplicar la metodologia de Groth-Sahai.

Ara té sentit parlar de l'algorisme de rastreig:

Rastreig(TK,σ,PP): L'algorisme de rastreig pren com a entrada la clau secreta TK, una firma σ i els paràmetres públics PP. Per rastrejar un missatge simplement calcula $Com(K)^{\delta_p} = K^{\delta_p}$, on $\delta_p = 1 \mod p$ i $\delta_p = 0 \mod q$. Amb això aconsegueix un valor no aleatoritzat que es pot relacionar amb un usuari interactuant amb l'entitat proveïdora d'atributs.

Teorema 2. *L'esquema modificat és totalment anònim i no enllaçable sempre i quan l'adversari no tingui accés a la factorització de n.*

Prova. El reptador pot usar l'algorisme d'**Inici** per començar el joc de seguretat. Posteriorment, com que coneix tots els paràmetres públics i secrets, pot usar els algorismes **GenClau, Firma, Verificació, Rastreig** per respondre a peticions i al repte.

El fet que l'avantatge de l'adversari en la sortida sigui negligible segueix directament del fet que σ_1 no revela informació sobre els atributs usats, doncs és com un element aleatori. De la mateixa manera, $\{(C_i, \pi_i)\}$ no revelen informació sobre f_i i finalment $Com(K), Com(\sigma_3), Com(\sigma_K), \pi_K$ i π_σ també són com elements aleatoris a la vista d'un adversari. □

Teorema 3. *L'esquema modificat és rastrejable sempre i quan l'adversari no tingui accés a la factorització de n ni a la clau MK.*

Prova. El reptador pot usar l'algorisme d'**Inici** per començar el joc de seguretat. Posteriorment, com que coneix tots els paràmetres públics i secrets, pot usar els algorismes **GenClau, Firma** per respondre a peticions.

El fet que l'esquema de firma automòrfic sigui segur implica que qualsevol firma ha d'utilitzar un valor de K que provingui d'una petició i, per tant, l'adversari no pot guanyar el joc de rastreig. És important remarcar que això es deu a la seguretat de la firma automòrfica sobre K. □

Teorema 4. *L'esquema modificat és no falsificable existencialment sota un atac de missatge escollit, en el model de l'oracle aleatori i assumint la hipòtesi del CDH en $\mathbb{G}_p$.*

Prova. En aquesta prova construïm un adversari $\mathcal{B}$ que soluciona el problema CDH en $\mathbb{G}_p$ interactuant amb un adversari $\mathcal{A}$ que trenca l'esquema ABS presentat. Cal notar que cada (C_i, π_i) ha de passar la primera equació de verificació, cosa que implica que C_i té la forma $C_i = (H_1(\mathsf{at}_i)/w)^{f_i} h^{z_i}$ per algun $f_i \in \{0, 1\}$ i $z_i \in \mathbb{Z}_n$. Distingim dos casos segons quin sigui el valor $\sum_{\mathsf{at}_i \in B} f_i$:

1. L'adversari del tipus-1 $\mathcal{A}_1$ és tal que fa una falsificació per un llindar l però la suma dels exponents f_i no suma l.

2. L'adversari del tipus-2 $\mathcal{A}_2$ és tal que fa una falsificació per un llindar l i la suma dels exponents f_i és l, per tant hi ha exactament l exponents f_i que són iguals a 1.

Per cada tipus d'adversari $\mathcal{A}_1$ i $\mathcal{A}_2$ construirem algorismes $\mathcal{B}_1$ i $\mathcal{B}_2$ per trencar el problema CDH en $\mathbb{G}_p$. La demostració surt dels dos lemes que venen a continuació.

Lema 1. *Si existeix un adversari del tipus-1 $\mathcal{A}_1$, aleshores existeix un algorisme $\mathcal{B}_1$, que trenca el problema CDH.*

Prova. Suposem que existeix un adversari del tipus-1 $\mathcal{A}_1$, que trenca la no-falsificació de l'esquema ABS proposat. L'algorisme $\mathcal{B}_1$ que trenca el problema CDH usant $\mathcal{A}_1$ rep: la descripció d'un grup bilinear $\mathbb{G}$, la factorització p, q de l'ordre n, un repte CDH aleatori $(g_p, g_p^{\alpha}, g_p^{\beta}) \in \mathbb{G}_p^3$, on g_p és un generador de $\mathbb{G}_p$. L'objectiu de $\mathcal{B}_1$ és calcular $g_p^{\alpha\beta}$. L'algorisme $\mathcal{B}_1$ interactua amb $\mathcal{A}_1$ de la següent manera:

Inici: $\mathcal{B}_1$ selecciona un generador $h \in \mathbb{G}_q$ i valors aleatoris $r_1 \in \mathbb{Z}_q^*, r_2, r_3 \in \mathbb{Z}_q$. També tria els paràmetres d'un esquema de firma automòrfic. Selecciona els paràmetres públics $PP = (n, \mathbb{G}, \mathbb{G}_T, e, g = g_p h^{r_1}, g_1 = g_p^{\alpha} h^{r_2}, h, h_1 = h^{r_2/r_1}, w = g_p^{\beta} h^{r_3}, H_1, H_2, PK_{\sigma})$ i dona PP a $\mathcal{A}_1$. Els paràmetres públics estan ben distribuïts ja que es compleix $e(g_1, h) = e(g_p^{\alpha} h^{r_2}, h) = e(h^{r_1}, h^{r_2/r_1}) = e(g_p h^{r_1}, h^{r_2/r_1}) = e(g, h_1)$, conseqüència de tenir $e(g_p^{\alpha}, h) = 1$ ja que tenim $g_p \in \mathbb{G}_p$ i $h \in \mathbb{G}_q$.

Peticions: Adaptativament, $\mathcal{A}_1$ pot anar fent peticions del hash H_1, del hash H_2, de clau privada o de firma. Per les peticions a un hash, $\mathcal{B}_1$ manté una llista (per cada hash) per tal de ser consistent. També es guarda una llista d'elements K usats, juntament amb els valors necessaris per calcular K.

Per una petició al hash H_1 en un atribut at_i, $\mathcal{B}_1$ genera un aleatori $c_i \in \mathbb{Z}_n$ i respon amb $H_1(\mathsf{at}_i) = g^{c_i}$. Per una petició pel hash H_2 per un missatge M, un llindar l i atributs B, $\mathcal{B}_1$ genera un aleatori $d_i \in \mathbb{Z}_n$ i respon amb $H_2(M_i, l, B) = g^{d_i}$. Per una petició per una clau privada per un conjunt d'atributs A, $\mathcal{B}_1$ genera i es guarda un element aleatori $e_j \in \mathbb{Z}_n$, calcula $K = g^{e_j}$, calcula la firma σ_K i per cada $\mathsf{at}_i \in A$ calcula $SK_i = (g_1^{c_i} K^{r_i}, g^{r_i})$. Respon amb $SK_A = (K, \{SK_i\}_{\mathsf{at}_i \in A})$. Fent servir les claus privades és senzill respondre a peticions de firma: primer es crea una clau secreta i després es realitza la firma.

Sortida: Finalment, $\mathcal{A}_1$ treu una firma falsificada $(\sigma^*, M^*, l^*, B^*)$ on

$$\sigma^* = (\sigma_1, \sigma_2, Com(\sigma_3), \{(C_i, \pi_i)\}_{\mathsf{at}_i \in B}, Com(K), Com(\sigma_K), \pi_K, \pi_\sigma)$$

Si (1) $\mathcal{A}_1$ ha fet alguna petició per una clau privada tal que el conjunt d'atributs A de la petició sigui tal que el conjunt $A \cap B^*$ té cardinal l^*, o (2) si $\mathcal{A}_1$ ha demanat una firma pel missatge M^*, el conjunt d'atributs B^* i el llindar l^* o (3) si l'equació de verificació no es compleix, aleshores $\mathcal{B}_1$ para la simulació doncs $\mathcal{A}_1$ no ha aconseguit una falsificació.

Altrament, $\mathcal{B}_1$ trenca el problema CDH de la següent manera: sigui δ_p tal que $\delta_p = 0 \mod q$ i $\delta_p = 1 \mod p$. Com que $u^{\delta_p} = 1$ si, i només si, $u \in \mathbb{G}_q$, aleshores tenim

$u^{\delta_p} \in \mathbb{G}_p$. Obtenim $C_i^{\delta_p} = (H_1(\mathsf{at}_i)^{\delta_p}/w^{\delta_p})^{f_i} = (g_p^{c_i}/g_p^{\beta})^{f_i}$ per a $\mathsf{at}_i \in B$ i, per tant, $C^{\delta_p} = \prod_{\mathsf{at}_i \in B} C_i^{\delta_p} = g_p^c/(g_p^{\beta})^f$, on $c = \sum_{\mathsf{at}_i \in B} c_i f_i$ i $f = \sum_{\mathsf{at}_i \in B} f_i$. De la segona equació de verificació modificada (secció 4.4.1) obtenim $e(g_p, \sigma_1^{\delta_p}) = e(g_p^{\alpha}, (g_p^{\beta})^{l^*} g_p^c/(g_p^{\beta})^f) e(\sigma_2^{\delta_p}, g_p^d) e(\sigma_3^{\delta_p}, g_p^e)$, on $H_2(M^*, l^*, B^*)^{\delta_p} = g_p^d$ i $K^{\delta_p} = g_p^e$. Aquesta equació que ve del fet que $Com(\sigma_3)^{\delta_p} = \sigma_3^{\delta_p}$, $Com(K)^{\delta_p} = K^{\delta_p}$ i $e(h, \pi_\sigma)^{\delta_p} = 1$. Si re-escrivim l'equació, obtenim

$$e(g_p^{\alpha}, g_p^{\beta})^{l^*-f} = e(g_p, \sigma_1^{\delta_p} (\sigma_2^{\delta_p})^{-d} (\sigma_3^{\delta_p})^{-e} (g_p^{\alpha})^{-c})$$

$\mathcal{B}_1$ pot recuperar els c_i corresponents als atributs de B, d corresponent a M^* gràcies a les llistes de cada hash. També pot recuperar el valor de e corresponent a K. Com que suposem que tenim un esquema de firma automòrfic segur, podem assegurar que el valor de K usat prové d'una petició i per tant és conegut. Per altra banda, $\mathcal{B}_1$ recupera $\{f_i\}_{i \in B}$ gràcies al fet que $C_i^{\delta_p} = 1$ si, i només si, $f_i = 0$. Com que suposavem $f = \sum_{\mathsf{at}_i \in B} f_i \neq l^*$, sabem que $(l^* - f)^{-1} \mod p$ existeix. Per tant, $\mathcal{B}_1$ trenca el problema CDH:

$$g_p^{\alpha\beta} = (\sigma_1^{\delta_p} (\sigma_2^{\delta_p})^{-d} (g_p^{\alpha})^{-c})^{1/(l^*-f)}.$$

Com que $\mathcal{B}_1$ té èxit cada cop que $\mathcal{A}_1$ té èxit, aleshores tenim $\mathrm{Adv}_{\mathcal{B}_1}^{\mathrm{CDH}} \geq \mathrm{Adv}_{\mathcal{A}_1}^{\mathrm{ABS}}$. □

Lema 2. *Si existeix un adversari del tipus-2 $\mathcal{A}_2$, aleshores existeix un algorisme $\mathcal{B}_2$, que trenca el problema CDH.*

Prova. Suposem que existeix un adversari del tipus-2 $\mathcal{A}_2$, que trenca la no-falsificació de l'esquema ABS proposat. L'algorisme $\mathcal{B}_2$ que trenca el problema CDH usant $\mathcal{A}_2$ rep: la descripció d'un grup bilinear $\mathbb{G}$, la factorització p, q de l'ordre n, un repte CDH aleatori $(g_p, g_p^{\alpha}, g_p^{\beta}) \in \mathbb{G}_p^3$, on g_p és un generador de $\mathbb{G}_p$. L'objectiu de $\mathcal{B}_1$ és calcular $g_p^{\alpha\beta}$. L'algorisme $\mathcal{B}_2$ interactua amb $\mathcal{A}_2$ de la següent manera:

Inici: $\mathcal{B}_2$ selecciona un generador $h \in \mathbb{G}_q$ i valors aleatoris $r_1 \in \mathbb{Z}_q^*, r_2, r_3, r_4 \in \mathbb{Z}_q, r_5 \in \mathbb{Z}_p$. Escull els paràmetres de la firma automòrfica i selecciona els paràmetres públics $PP = (n, \mathbb{G}, \mathbb{G}_T, e, g = g_p h^{r_1}, g_1 = g_p^{\alpha} h^{r_2}, h, h_1 = h^{r_2/r_1}, w = g_p^{r_5} h^{r_3}, H_1, H_2, PK_\sigma)$, defineix $g_2 = g_p^{\beta} h^{r_4}$ i dona PP a $\mathcal{A}_2$. Els paràmetres públics estan ben distribuïts ja que $e(g_1, h) = e(g_p^{\alpha} h^{r_2}, h) = e(h^{r_1}, h^{r_2/r_1}) = e(g_p h^{r_1}, h^{r_2/r_1}) = e(g, h_1)$. Amb aquests paràmetres queda definit implícitament $s = \log_g g_1$.

Peticions: Adaptativament, $\mathcal{A}_2$ pot anar fent peticions del hash H_1, del hash H_2, de clau privada o de firma. Per les peticions al hash, $\mathcal{B}_2$ manté una llista (per cada hash) per tal de ser consistent. També es guarda una llista d'elements K usats, juntament amb els valors necessaris per calcular K.

Per una petició al hash H_1 en un atribut at_i, $\mathcal{B}_2$ respon de la següent manera: si ja s'havia fet una petició al hash H_1 pel mateix atribut, recupera $(\mathsf{at}_i, H_1\text{-}moneda_i, c_i)$ de la llista del hash H_1. Altrament, genera una $H_1\text{-}moneda_i \in \{0, 1\}$ de manera que $\Pr[H_1\text{-}moneda_i = 1] = \rho_1$, per un ρ_1 que determinarem després. $\mathcal{B}_2$ genera també un aleatori

$c_i \in \mathbb{Z}_n$ i registra $(\mathsf{at}_i, H_1\text{-}moneda_i, c_i)$ en la llista del hash H_1. Si $H_1\text{-}moneda_i = 0$, aleshores respon amb $H_1(\mathsf{at}_i) = g^{c_i}$, altrament respon amb $H_1(\mathsf{at}_i) = g_2^{c_i}$.

Per una petició pel hash H_2 per un missatge M_k, llindar l_k i conjunt d'atributs B_k, $\mathcal{B}_2$ respon de la següent manera: si ja s'havia fet una petició al hash H_2 pel mateix missatge, llindar i conjunt d'atributs, recupera (M_k, l_k, B_k, d_k) de la llista del hash H_2. Altrament, genera un aleatori $d_k \in \mathbb{Z}_n$, registra (M_k, l_k, B_k, d_k) en la llista del hash H_2 i respon amb $H_2(M_k, l_k, B_k) = g^{d_k}$.

Per una petició de clau privada per un conjunt d'atributs A, $\mathcal{B}_2$ respon de la següent manera: genera una $K\text{-}moneda_j \in \{0,1\}$ de manera que $\Pr[K\text{-}moneda_j = 1] = \rho_2$, per un ρ_2 determinat després. Si $K\text{-}moneda_j = 0$, $\mathcal{B}_2$ genera un aleatori $e_j \in \mathbb{Z}_n$ i fixa $e'_j = 0$, altrament genera uns aleatoris $e_j, e'_j \in \mathbb{Z}_n^*$. Registra $(K\text{-}moneda_j, e_j, e'_j)$ en la llista de identitats. Defineix $K_j = g^{e_j} g_1^{e'_j}$. Posteriorment, per cada atribut de A, recupera de la llista de hash-1 $(\mathsf{at}_i, H_1\text{-}moneda_i, c_i)$. Si $H_1\text{-}moneda_i = 0$ escull un nombre aleatori $r_i \in \mathbb{Z}_n$ i defineix $SK_{j,i} = (g_1^{c_i} K_j^{r_i}, g^{r_i})$. Altrament, si $H_1\text{-}moneda_i = 1$ i $K\text{-}moneda_j = 1$ genera un aleatori $r_i \in \mathbb{Z}_n$ i defineix $SK_{j,i} = ((g_2^{c_i})^{-\frac{e_j}{e'_j}} (g^{e_j} g_1^{e'_j})^{r_i}, g^{r_i} g_2^{-\frac{c_i}{e'_j}})$. La clau secreta està ben distribuïda (denotem β^* el valor $\log_g g_2$): $E_{j,i} = (g_2^{c_i})^{-\frac{e_j}{e'_j}} (g^{e_j} g_1^{e'_j})^{r_i} = (g_2^{c_i})^s (g^{e_j} g_1^{e'_j})^{(r_i - \frac{c_i}{e'_j}\beta^*)} = H(\mathsf{at}_i)^s K_j^{r'_i}$ i, per altra banda, $G_{j,i} = g^{r_i} g_2^{-\frac{c_i}{e'_j}} = g^{(r_i - \frac{c_i}{e'_j}\beta^*)} = g^{r'}$. Finalment, en el cas que $H_1\text{-}moneda_i = 1$ i $K\text{-}moneda_j = 0$, aleshores $\mathcal{B}_2$ no pot crear $SK_{j,i}$. Si $\mathcal{B}_2$ pot crear $SK_{j,i}$ per tot $\mathsf{at}_i \in A$ aleshores calcula la firma sobre K i respon amb $SK_A = (K_j, \sigma_K, \{SK_{j,i}\}_{\mathsf{at}_i \in A})$. Altrament, aborta. En la figura 4.1 es pot veure un esquema de les situacions en les que el reptador abortarà en una petició de clau privada.

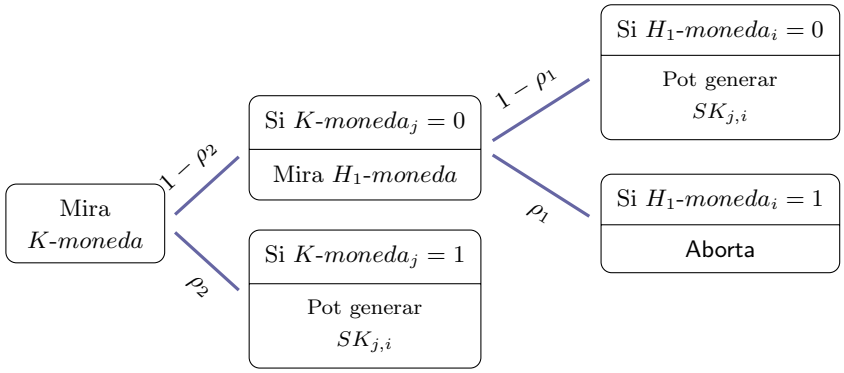


Figura 4.1: *Casos possibles en una petició per clau secreta*

Per una petició de firma amb un conjunt d'atributs possibles B, un llindar l i un missatge M, $\mathcal{B}_2$ respon de la següent manera: primer de tot es recupera de les llistes la següent informació: (M_k, l_k, B_k, d_k) i $(H(\mathsf{at}_i), H_1\text{-}moneda_i, c_i)$ per tots els atributs en B. Crea un valor $K = g^e g_1^{e'}$. Donat que amb aquest valor de K pot crear claus secretes per qualsevol atribut at, $\mathcal{B}_2$ genera una clau secreta SK_B i firma el missatge M de la mateixa

manera que ho faria un usuari qualsevol. Cal remarcar que l'adversari no pot distingir quin valor de K s'ha usat en la firma.

Sortida: Finalment, $\mathcal{A}_2$ treu una firma falsificada $(\sigma^*, M^*, l^*, B^*)$ on

$$\sigma^* = (\sigma_1, \sigma_2, Com(\sigma_3), \{(C_i, \pi_i)\}_{\mathsf{at}_i \in B^*}, Com(K^*), Com(\sigma_K^*), \pi_K^*, \pi_\sigma^*).$$

Si (1) s'ha fet una petició per clau privada per algun conjunt d'atributs A tal que el conjunt $A \cap B^*$ té cardinal l^*, o (2) si $\mathcal{A}_2$ ha demanat una firma pel missatge M^*, el conjunt d'atributs B^* i el llindar l^* o (3) si l'equació de verificació no es compleix, aleshores $\mathcal{B}_2$ para la simulació doncs $\mathcal{A}_2$ no ha aconseguit una falsificació.

Altrament, $\mathcal{B}_2$ trenca el problema CDH de la següent manera: sigui δ_p tal que $\delta_p = 0 \mod q$ i $\delta_p = 1 \mod p$. Calculant $C_i^{\delta_p}$ per tot $\mathsf{at}_i \in B$, pot extreure els valors $\{f_i\}_{\mathsf{at}_i \in B}$, recuperant el conjunt A_S d'atributs usats en la falsificació. Aleshores recupera (M^*, l^*, B^*, d^*) i $(\mathsf{at}_{i^*}, H_1\text{-}moneda_{i^*}, c_{i^*})$ per tots els atributs en A. Per altra banda, calcula $Com(K^{\delta_p}) = K^{\delta_p}$. Donat que l'esquema de firma automòrfic és segur, el fet que la firma sigui vàlida implica que el valor de K usat i la seva firma hagin estat obtinguts d'una petició de clau secreta. Per tant obté K i recupera $(K\text{-}moneda_*, e^*, (e^*)')$. En cas que $K\text{-}moneda_* = 1$, aleshores $\mathcal{B}_2$ aborta. Altrament, anomenem B_0 el conjunt d'índexs i^* tal que $H_1\text{-}moneda_{i^*} = 0$ i B_1 el conjunt d'índexs i^* tal que $H_1\text{-}moneda_{i^*} = 1$. (A partir d'ara, obviarem els asteriscs usant com a exponents c_i, d, e).Per la no mal·leabilitat del hash, σ_1 ha de ser de la forma

$$\begin{aligned}
\sigma_1 &= \left(\prod_{\mathsf{at}_i \in A} (H(\mathsf{at}_i))^s \right) K^r H_m^t h_1^z \\
&= \prod_{i \in B_0} (H(\mathsf{at}_i))^s \prod_{i \in B_1} (H(\mathsf{at}_i))^s (g^e)^r (g^d)^t h_1^z \\
&= \prod_{i \in B_0} (g^{c_i})^s \prod_{i \in B_1} (g_2^{c_i})^s \sigma_3^e \sigma_2^d h_1^z \\
&= g_1^{\sum_{i \in B_0} c_i} g_2^{s \sum_{i \in B_1} c_i} \sigma_3^e \sigma_2^d h_1^z
\end{aligned}$$

Denotem $c_{B_0} = \sum_{i \in B_0} c_i$ i $c_{B_1} = \sum_{i \in B_1} c_i$. Si $c_{B_1} \bmod p = 0$, aleshores aborta ja que no pot solucionar el problema CDH. Altrament, obtenim $g_2^s = (\sigma_1 \sigma_2^{-d} \sigma_3^{-e} g_1^{-c_{B_0}} h_1^{-z})^{1/c_{B_1}}$. Per altra banda, tenim $(g_2^s)^{\delta_p} = g_p^{\alpha\beta}$, que ve del fet que $g_p^\alpha = g_1^{\delta_p} = (g^s)^{\delta_p}$. Usant que $Com(\sigma_3)^{\delta_p} = \sigma_3^{\delta_p}$, es soluciona el problema CDH calculant:

$$g_p^{\alpha\beta} = (g_2^s)^{\delta_p} = (\sigma_1^{\delta_p} (\sigma_2^{\delta_p})^{-d} (\sigma_3^{\delta_p})^{-e} (g_p^\alpha)^{-c_{B_0}})^{1/c_{B_1}}$$

En la figura 4.2 es pot veure un esquema de les situacions en les que el reptador abortarà després de la sortida de l'adversari.

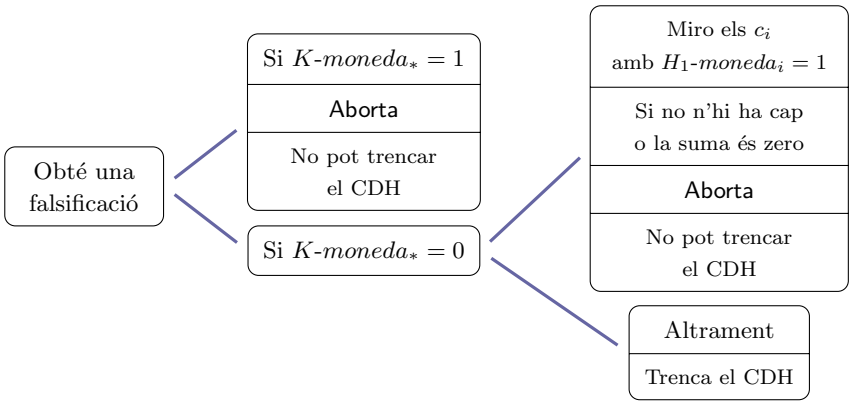


Figura 4.2: *Possibles valors dels paràmetres en la falsificació segons si s'aborta o no*

Anàlisi: Per l'anàlisi, sigui aborta l'event en que $\mathcal{B}_2$ aborta durant la simulació i $\mathsf{falsifica}$ l'event en que $\mathcal{A}_2$ produeixi una falsificació vàlida d'acord amb la definició del joc de falsificació. Tenim

$$\begin{aligned}\mathrm{Adv}^{\mathrm{CDH}}_{\mathcal{B}_2} &\geq Pr[\mathsf{falsifica} \wedge \neg\mathsf{aborta}] \\ &= Pr[\mathsf{falsifica}|\neg\mathsf{aborta}]Pr[\neg\mathsf{aborta}] \\ &= \mathrm{Adv}^{\mathrm{ABS}}_{\mathcal{A}_2} Pr[\neg\mathsf{aborta}]\end{aligned}$$

La tercera igualtat surt del fet que si no s'aborta, aleshores la simulació és equivalent al joc de no falsificació. Ara definim aborta_E l'event que $\mathcal{B}_2$ aborti quan li demanen una clau privada, aborta_K l'event K-$moneda_* = 1$ i aborta_C l'event $c_{B_1} \mod p = 0$. Tenim

$$\begin{aligned}Pr[\neg\mathsf{aborta}] &= Pr[\neg\mathsf{aborta}_E \wedge \neg\mathsf{aborta}_K \wedge \neg\mathsf{aborta}_C] \\ &= Pr[\neg\mathsf{aborta}_E]Pr[\neg\mathsf{aborta}_K \wedge \neg\mathsf{aborta}_C|\neg\mathsf{aborta}_E] \\ &= Pr[\neg\mathsf{aborta}_E]Pr[\neg\mathsf{aborta}_K|\neg\mathsf{aborta}_E]Pr[\neg\mathsf{aborta}_C|\neg\mathsf{aborta}_E] \\ &\geq \Big[((1-\rho_2)(1-\rho_1)^{q_E} + \rho_2)\Big] \cdot \Big[(1-\rho_2)(1-\rho_1)^{q_E}\Big] \cdot \\ &\quad \cdot \Big[(1-1/p)(1-(1-\rho_1)^{q_E})\rho_2\Big]\end{aligned}$$

La tercera igualtat segueix del fet que els events aborta_K i aborta_C són independents entre sí. La desigualtat surt del càlcul de probabilitats: el primer factor surt de (1) considerar la probabilitat de no abortar en una petició per clau privada, (2) fitar la probabilitat de no abortar en totes les peticions per clau privada, (3) usar els multiplicadors de Lagrange per veure el nombre òptim d'atributs en cada petició, que és que totes les peticions tinguin el mateix nombre d'atributs i (4) optimitzar el nombre de claus secretes

demanades tenint en compte que el nombre total d'atributs és fix. Els altres dos factors surten directament del càlcul de probabilitats i d'aplicar certes fites. Per simplificar el càlcul, suposem que q_E correspon al nombre total d'atributs demanats, que està fixat.

Finalment, cal notar que $f(\rho_1, \rho_2) = ((1-\rho_2)(1-\rho_1)^{q_E} + \rho_2)(1-\rho_2)(1-\rho_1)^{q_E}(1 - 1/p)(1-(1-\rho_1)^{q_E})\rho_2$ és major que 0 excepte si $\rho_1 = 0, \rho_1 = 1, \rho_2 = 0$ o $\rho_2 = 1$. Escollint valors per ρ_1 i ρ_2 (excepte els casos esmentats) obtenim que

$$\mathrm{Adv}^{\mathrm{CDH}}_{\mathcal{B}_2} \geq \mathrm{Adv}^{\mathrm{ABS}}_{\mathcal{A}_2} \cdot \Omega(1)$$

□

4.5 Aconseguint estructures d'accés genèriques

L'esquema anterior només permet fer estructures de llindar. En aquest apartat proposem una ampliació de l'esquema ABS per estructures d'accés de $\mathbb{Z}_n$-mòdul. Abans, però, fem un esbós de com aconseguir estructures més senzilles que les estructures d'accés de $\mathbb{Z}_n$-mòdul però sense perdre eficiència respecte l'esquema inicial.

4.5.1 Estructures multi-llindar

Definim una estructura d'accés bi-llindar com una estructura on dividim els atributs en dos conjunts B_1, B_2 i una col·lecció d'atributs és autoritzada si, i només si, el nombre d'atributs de la col·lecció dins B_1 és igual a un llindar l_1 i el nombre d'atributs en B_2 és igual a un llindar l_2. De la mateixa manera, podem estendre aquest concepte a estructures multi-llindar. Cal notar que les estructures multi-llindar són un cas particular de les estructures d'accés multi-partites.

Per realitzar aquestes estructures n'hi ha prou amb definir w_j per cada conjunt B_j enlloc de tenir un únic valor w. Ara, els compromisos de les claus públiques seran $C_i = (H_1(\mathsf{at}_i)/w_j)^{f_i} h^{z_i}$ si at_i pertany a B_j. Les proves NIWI π_i es canviaran de la mateixa manera.

En el cas bi-llindar, la segona equació de verificació (la que involucra tots els valors C_i), queda de la següent manera:

$$e(\sigma_1, g) \stackrel{?}{=} e(w_1^{l_1} \prod_{\mathsf{at}_i \in B_1} C_i, g_1) e(w_2^{l_2} \prod_{\mathsf{at}_i \in B_2} C_i, g_1) e(H_m, \sigma_2) e(Com(K), Com(\sigma_3)) e(h, \pi_\sigma)$$

La demostració de no falsificació és molt semblant a la ja presentada. Per una banda, en el cas de l'adversari tipus-2 la demostració és totalment idèntica. En el cas d'un adversari tipus-1 podem tractar el cas en que el nombre d'atributs de B_j usats en la firma no correspongui a l_j per cada conjunt definit i modificar lleument la demostració.

4.5.2 Estructures de $\mathbb{Z}_n$-mòdul

Una estructura d'accés Γ és realitzable per un esquema de compartició de secrets de $\mathbb{Z}_n$-mòdul si existeix un enter positiu h i una aplicació $\psi : \mathcal{P} \cup \{D\} \to (\mathbb{Z}_n)^h$ tal que $A \in \Gamma$ si, i només si, $\psi(D) \in \langle \psi(R_i) \rangle_{R_i \in A}$.

Normalment, si un distribuïdor vol repartir un valor secret $s \in \mathbb{Z}_n$ d'acord amb l'estructura d'accés, pren un vector aleatori $w \in (\mathbb{Z}_n)^h$ tal que $w \cdot \psi(D) = s$. El fragment de l'element $R_i \in \mathcal{P}$ és $s_i = w \cdot \psi(R_i) \in \mathbb{Z}_n$. Sigui A un subconjunt autoritzat, $A \in \Gamma$, aleshores, per definició, $\psi(D) = \sum_{R_i \in A} \lambda_i^A \psi(R_i)$, per alguns valors $\lambda_i^A \in \mathbb{Z}_n$. Per recuperar el secret a partir dels seus fragments, es calcula

$$\sum_{R_i \in A} \lambda_i^A s_i = \sum_{R_i \in A} \lambda_i^A (w \cdot \psi(R_i)) = w \cdot \sum_{R_i \in A} \lambda_i^A \psi(R_i) = w \cdot \psi(D) = s$$

En el nostre esquema usarem l'estructura de $\mathbb{Z}_n$-mòdul d'una manera diferent. Denotem per B el conjunt d'atributs que apareixen en Γ, és a dir, $B = \{\mathsf{at}_i | \exists\ A \text{ tq } \mathsf{at}_i \in A, A \in \Gamma\}$. B és el conjunt $\mathcal{P}$ de l'aplicació i els seus elements són els atributs at_i. La idea és la següent: el firmant ha de convèncer al verificador de que té un conjunt A d'atributs autoritzats, és a dir, que existeixen uns valors λ_i^A tals que $\sum_{at_i \in A} \lambda_i^A \psi(\mathsf{at}_i) = \psi(D)$. Per preservar l'anonimat dels atributs, el firmant farà compromisos d'els valors λ_i per tot $\mathsf{at}_i \in B$ de la següent manera: si $\mathsf{at}_i \in A$, aleshores farà un compromís del valor $\lambda_i = \lambda_i^A$ mentre que si $\mathsf{at}_i \notin A$ el compromís serà del valor $\lambda_i = 0$. A més a més, la firma inclourà Proves Sense Coneixement per convèncer al verificador d'aquest fet.

Els valors de λ_i es relacionen amb els valors de C_i de la següent manera: $\lambda_i \neq 0$ si, i només si, $f_i \neq 0$. Això ho podem fer sense pèrdua de generalitat, doncs si algun valor λ_i^A és zero, podem excloure l'atribut corresponent de A. El firmant haurà de convèncer al verificador de la relació entre C_i i els compromisos de λ_i mitjançant Proves Sense Coneixement.

La manera que he trobat de fer-ho és la següent: n'hi ha prou amb demostrar $\lambda_i = \tilde{\lambda_i} f_i$ amb $\tilde{\lambda_i} \neq 0$. És clar que si $f_i = 1$ podem prendre $\tilde{\lambda_i} = \lambda_i$ i l'equació es compleix. Per altra banda, si $f_i = \lambda_i = 0$ podem prendre un $\tilde{\lambda_i} \neq 0$ a l'atzar, i l'equació es segueix complint.

Finalment, falta comprovar que $\tilde{\lambda_i} \neq 0\ \forall\ i \in A$. Per fer això, demostrem que cada $\tilde{\lambda_i}$ és invertible (i per tant diferent de zero): existeix μ_i tal que $\mu_i \tilde{\lambda_i} = 1\ \forall\ i \in A$. La probabilitat que $\tilde{\lambda_i}$ sigui diferent de zero però no sigui invertible és molt petita ja que n és el producte de dos nombres primers grans. Podríem pensar que una alternativa més eficient és comprovar que el producte de tots els valors és invertible però això no és possible, ja que les proves de Groth-Sahai només ens permeten demostrar equacions quadràtiques.

En resum, donada l'estructura d'accés definida per la funció ψ i els compromisos del hash dels atributs $C_i = (H_1(\mathsf{at}_i))^{f_i} h^{z_i}$ (que denoten els atributs usats en la firma), hem de fer proves de Groth-Sahai per demostrar que:

1. $\exists\vec{\lambda}$ tq $\Psi\vec{\lambda} = \psi(\mathcal{D})$ o, el que és el mateix, $\Psi_j\vec{\lambda} = \psi_j(\mathcal{D})$ per $j = \{1, \ldots, h\}$.
2. Si denotem $\vec{\lambda} = (\lambda_1, \ldots, \lambda_a)$, demostrar que $\exists\tilde{\lambda_i}$ tq $\lambda_i = \tilde{\lambda_i} f_i$ per $i = \{1, \ldots, a\}$.
3. Demostrar que per tot $i = \{1, \ldots, a\}$, $\exists\mu_i$ tq $\mu_i\tilde{\lambda_i} = 1$.

on Ψ és la matriu de la aplicació ψ: cada columna representa $\psi(\mathsf{at}_i)$. A més usem Ψ_j per denotar la fila j-éssima de Ψ.

Finalment, com que f_i no ve d'un compromís 'dels habituals', hem de fer un compromís de f_i i demostrar que C_i i aquest compromís comprometen el mateix valor. L'equació a demostrar és: $e(H_1(\mathsf{at}_i)^{f_i}, g) \cdot e(g^{f_i}, H_1(\mathsf{at}_i)^{-1}) = 1$, on les variables són $H_1(\mathsf{at}_i)^{f_i}$ i g^{f_i}, que comprometem C_i i a un compromís de l'exponent Groth-Sahai, respectivament.

L'esquema resultant és el següent:

Inici(λ): L'algorisme d'inicialització és el mateix que en el cas llindar amb la diferència que no s'escull w.

GenClau(A,MK,PP): L'algorisme de generació de claus és el mateix que en el cas llindar.

Firma(M, Γ, SK_A, PP) L'algorisme de firma pren com a entrada un missatge M, una estructura d'accés Γ i la clau privada de l'usuari SK_A. Denotem per B el conjunt d'atributs que apareixen en Γ, és a dir, $B = \{\mathsf{at}_i | \exists\ A \text{ tq } \mathsf{at}_i \in A, A \in \Gamma\}$. L'usuari selecciona un conjunt d'atributs autoritzats $A_S \in \Gamma$ i que posseeixi, és a dir, $A_S \subset A$. Per una banda, per tots els atributs de B, calcula

$$C_i = (H_1(\mathsf{at}_i))^{f_i} h^{z_i} \text{ i } \pi_i = ((H_1(\mathsf{at}_i))^{2f_i - 1} h^{z_i})^{z_i}$$

on $f_i = 1$ si $\mathsf{at}_i \in A$ i $f_i = 0$ altrament.

A més a més, per tots els atributs de B, calcula els següents compromisos:

$$Com(f_i), Com(\tilde{\lambda_i}), Com(\lambda_i)$$

i les proves NIWI de que els valors compromesos compleixen les equacions anteriors: $\pi_{f_i}, \pi_{\lambda_i}, \pi_{\mu_i}$ per tot $\mathsf{at}_i \in B$ i π_{ψ_k} per tot $k \in \{1, \ldots, h\}$.

Per altra banda calcula $H_m = H_2(M, \Gamma)$, escull un nombre aleatori $t \in \mathbb{Z}_n$ i calcula

$$\sigma_1 = \left(\prod_{\mathsf{at}_i \in A} E_i \right) H_m^t h_1^z \ , \ \sigma_2 = g^t \text{ i } \sigma_3 = \prod_{\mathsf{at}_i \in A} G_i.$$

on $z = \sum_{\mathsf{at}_i \in B} z_i$. Calcula el compromís de σ_3 i a K i la prova NIWI de que són valors que compleixen l'equació de verificació. Finalment, la firma és:

$$\begin{aligned} \sigma = (&\{C_i, \pi_i, Com(f_i), Com(\tilde{\lambda_i}), Com(\lambda_i), \pi_{f_i}, \pi_{\lambda_i}, \pi_{\mu_i}\}_{\mathsf{at}_i \in B}, \\ &\{\pi_{\psi_k}\}_{k=1}^{h}, \sigma_1, \sigma_2, Com(\sigma_3), Com(K), Com(\sigma_K), \pi_K, \pi_\sigma) \end{aligned}$$

Verificació (σ, Γ, PP): L'algorisme de verificació pren com a entrada una firma σ, el missatge M i l'estructura d'accés Γ. Procedeix de la següent manera:

1. Per tot $\mathsf{at}_i \in B$ comprova si es compleix $e(C_i, C_i/(H_1(\mathsf{at}_i))) \stackrel{?}{=} e(h, \pi)$.
2. Per tot $\mathsf{at}_i \in B$, comprova que es compleix $e(C_i, g) \cdot e(Com(f_i), H(\mathsf{at}_i)^{-1}) = e(h, \pi_{f_i})$
3. Per tot $\mathsf{at}_i \in B$, comprova que es compleix $e(Com(\lambda_i), g)^{-1} \cdot e(Com(\tilde{\lambda_i}), Com(f_i)) = e(h, \pi_{\lambda_i})$
4. Per tot $\mathsf{at}_i \in B$, comprova que es compleix $e(Com(\lambda_i), Com(\mu_i)) = e(h, \pi_{\mu_i})$.
5. Per tot j des d'1 fins a h, comprova que es compleix $\prod_{\mathsf{at}_i \in B} e(Com(\lambda_i), g^{\psi^{(j)}(\mathsf{at}_i)}) = e(h, \pi_{\psi_j})$
6. Usant la verificació d'una prova de Groth-Sahai, comprova que la firma de K (compromesa) és vàlida.
7. Comprova si es compleix $e(\sigma_1, g) \stackrel{?}{=} e(\prod_{\mathsf{at}_i \in B} C_i, g_1) e(H_m, \sigma_2) e(Com(K), Com(\sigma_3)) e(h, \pi_\sigma)$
8. Si tots els passos són correctes, la firma és vàlida, altrament no ho és.

Teorema 5. *L'esquema és correcte, anònim, no enllaçable i no falsificable.*

Prova La correcció surt directe de desenvolupar les equacions de verificació. L'anonimat i no enllaçabilitat surt de veure que tots els elements que no depenen del missatge no revelen informació sobre els atributs o la identitat del signant. Finalment, la no falsificabilitat surt del Lema 2, amb petites modificacions. □

4.6 ABS en el model estàndard

En aquesta secció usem diferents metodologies per transformar l'esquema ABS anterior de manera que la seguretat estigui basada en el model estàndard, enlloc del model de l'oracle aleatori.

Per una banda tenim el hash H_1, que s'usa per transformar atributs (cadenes de caràcters) a elements de G. Una possibilitat per evitar el hash és reduir el conjunt d'atributs a una família designada en el procés d'inici. És a dir, per cada atribut at_i es genera un element $Q_i \in \mathbb{G}$ que l'identifica únicament i que està en la descripció dels paràmetres. Tot el fet abans segueix igual, tenint en compte que ara en les demostracions de seguretat es definirà $Q_i = g^{c_i}$ (o bé $Q_i = g_2^{c_i}$, segons convingui) per algun c_i. D'alguna manera, substituïm el hash H_1 per una assignació entre atributs i elements Q_i.

Per altra banda tenim el hash H_2, que s'usa per transformar missatges (cadenes de caràcters) a elements de G. La solució en aquest cas és agafar un hash resistent a col·lisions $H : \{0,1\}^* \to \{0,1\}^m$ i elements $v_1, v_2, \ldots, v_m \in \mathbb{G}$ i substituir el valor de H_2 per $\prod_{j=1}^{m} v_j^{H(M,\Gamma)_j}$ on $H(M,\Gamma)_j$ és la coordenada j-éssima de $H(M,\Gamma)$. En la demostració de seguretat s'agafaran els elements $v_1, v_2, \ldots, v_m$ coneixent-ne el logaritme discret amb base g. D'aquesta manera donat un missatge l'element que es firma serà g^d per algun d conegut per $\mathcal{B}$, que recordem que és qui intenta trencar el problema CDH. D'aquesta manera la demostració segueix sent vàlida.

4.7 Altres consideracions

4.7.1 Sobre les firmes automòrfiques

En l'esquema ABS requerim l'existència d'una firma automòrfica. Ara per ara, l'única particularització que se'n coneix és una deguda a Fuchsbauer [Fuc09]. En aquest article, Fuchsbauer presenta firmes automòrfiques per grups bilineals simètrics i per grups bilineals no simètrics. En el cas simètric, que és l'únic aplicable al nostre escenari, es basa en les hipòtesis *q-DHSDH* (en anglès, q-Double Hidden Symmetric Diffie-Hellman) i *WFCDH* (en anglès, Weak Flexible Computational Diffie-Hellman).

Aquestes hipòtesis, tot i no ser estàndards, són bastant raonables: sota la hipòtesi del coneixement de l'exponent, la primera és equivalent a la hipòtesi *q-SDH-III* que és una hipòtesi una mica més dèbil que *q-SDH*, una hipòtesi bastant estàndard. Per altra banda, la versió asimètrica del *WFCDH* és equivalent a la hipòtesi del logaritme discret sota la hipòtesi del coneixement de l'exponent.

Tot i que la hipòtesi necessita un grup bilineal d'ordre primer, això no és cap problema, doncs podríem treballar dins el subgrup d'ordre p de $\mathbb{G}$ o bé multiplicar els elements per elements d'ordre q de manera que segueixin complint les equacions pertinents.

4.7.2 Esquema ABS basat en altres hipòtesis

L'esquema presentat aquí treballa en grups d'ordre compost i es basa en la hipòtesi de la decisió de grup. És conegut que les operacions que es fan en grups d'ordre compost són molt ineficients. En aquesta secció pretenem donar alternatives.

Per una banda, es pot plantejar el mateix esquema sota la hipòtesis decisional lineal. Aquesta hipòtesis necessita un grup bilineal simètric d'ordre primer. El problema és que, mentre que en la hipòtesis de la decisió de grup el fet de comprometre valors i fer proves NIWI no augmenta el tamany de la firma, fer proves NIWI sota la hipòtesi decisional lineal implica triplicar el nombre d'elements de la firma i multiplicar per nou el nombre d'equacions de pairing que s'han de comprovar. Tenint en compte que el tamany d'un

grup compost és aproximadament deu cops el tamany d'un grup d'ordre primer, el tamany de la firma en bits és aproximadament el mateix. Per altra banda, calcular pairings en grups d'ordre primer és més eficient que calcular pairings en grups d'ordre compost, tot i tenint en compte el tamany.

Encara més interessant seria treballar sobre grups lineals asimètrics ja que són els que més interès pràctic tenen. El problema és que els compromisos C_i, que nosaltres usem per comprometre la unitat o un valor determinat, són tant senzills gràcies a la simetria del grup bilineal.

4.7.3 Estructures d'accés d'espai vectorial i de $\mathbb{Z}_n$-mòdul

Actualment existeixen diferents articles on es proposen estructures d'accés d'espai vectorial. Les estructures d'accés d'espai vectorial permeten representar una gran varietat d'estructures d'accés. Per exemple, un esquema de llindar es pot representar amb una estructura d'accés d'espai vectorial.

En aquest article, el fet de treballar en un mòdul compost no permet parlar d'espais vectorials sinó de $\mathbb{Z}_n$-mòduls. Això no suposa cap restricció important ja que al ser n un producte de primers grans l'anell $\mathbb{Z}_n$ es comporta en la majoria de casos com un cos.

De totes maneres, en la secció anterior s'ha proposat una variació de l'esquema ABS que permet treballar en grups bilineals d'ordre primer i per tant sí que es podria parlar d'espais vectorials.

4.8 Conclusió

En aquest article hem construït un esquema ABS segur sota el model estàndard. Hem donat tres estructures d'accés: estructura llindar, estructura multi-llindar i estructura de $\mathbb{Z}_n$-mòdul. Creiem que l'estratègia de fer servir proves de Groth-Sahai per garantir anonimat és molt poderosa però en el nostre cas limita l'eficiència i el tamany de la firma.

Encara falta, per tant, trobar un esquema ABS eficients amb estructures d'accés generals sobre grups bilineals asimètrics. Tot i això, creiem que la metodologia usada per trobar l'esquema és força interessant, ja que és modular (podem separar l'estructura d'accés de la firma pròpiament) i senzilla.

Conclusions

En aquest treball es proposa un nou esquema de firma digital, substancialment millor que els existents. Aquest és un resultat important: d'entrada, millora la seguretat dels esquemes anteriors notablement. L'esquema també presenta una gran flexibilitat pel que fa a les famílies d'atributs autoritzades per firmar, cosa que no es permet en d'altres esquemes per la manera com estan construïts.

Deixant de banda el resultat en sí, l'estudi de la metodologia de Groth-Sahai també és un punt molt important del treball. L'article de Groth i Sahai és força complicat i dens, per això crec necessari reflectir l'anàlisi de l'article en aquest treball. En la secció en què explico les proves de Groth-Sahai he intentat donar-hi tota la intuïció necessària de manera que es pugui entendre la metodologia amb més facilitat, però sense perdre la formalització que hi ha al darrera.

Com que Groth i Sahai presenten una metodologia totalment genèrica, crec que el potencial d'aquesta eina és molt gran. De fet, ja hi ha hagut diversos articles que les utilitzen constantment. En principi, l'aplicació més immediata és donar anonimat en diversos aspectes, però segurament tindrà més àmbits d'aplicació.

Quan vaig començar a estudiar les Proves Sense Coneixement em vaig trobar bastant perdut ja que tots els articles eren molt tècnics i amb notació complexa. És per això que he decidit que era important fer un apartat del treball que, començant des de zero, arribés fins a la formalització d'aquest concepte explicant-ne tota la intuïció.

Val a dir que la literatura de les Proves Sense Coneixement és molt extensa. Un exemple són les proves de coneixement, que podem considerar com un sistema de proves amb alguna propietat afegida. Aquestes proves són molt utilitzades en la criptografia, però la formalització és encara més complicada. Ara bé, un cop entès el marc general dels sistemes de proves, la comprensió dels diferents conceptes resulta més senzilla.

Finalment, l'estudi de la seguretat de les firmes digitals reflecteix tots els coneixements que he anat adquirint durant l'estudi de les eines criptogràfiques. Trobo que la comprensió de la manera de demostrar la seguretat en els esquemes criptogràfics és important per poder arribar a definir-ne de nous.

Bibliografia

[BBS04] Dan Boneh, Xavier Boyen, and Hovav Shacham. Short group signatures. In Matthew K. Franklin, editor, *CRYPTO*, volume 3152 of *Lecture Notes in Computer Science*, pages 41–55. Springer, 2004.

[BFM88] Manuel Blum, Paul Feldman, and Silvio Micali. Non-interactive zero-knowledge and its applications (extended abstract). In *STOC*, pages 103–112. ACM, 1988.

[BGN05] Dan Boneh, Eu-Jin Goh, and Kobbi Nissim. Evaluating 2-dnf formulas on ciphertexts. In Joe Kilian, editor, *TCC*, volume 3378 of *Lecture Notes in Computer Science*, pages 325–341. Springer, 2005.

[Boy07] Xavier Boyen. Mesh signatures. In Moni Naor, editor, *EUROCRYPT*, volume 4515 of *Lecture Notes in Computer Science*, pages 210–227. Springer, 2007.

[BR93] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *ACM Conference on Computer and Communications Security*, pages 62–73, 1993.

[BW06] Xavier Boyen and Brent Waters. Compact group signatures without random oracles. In Serge Vaudenay, editor, *EUROCRYPT*, volume 4004 of *Lecture Notes in Computer Science*, pages 427–444. Springer, 2006.

[BW07] Xavier Boyen and Brent Waters. Full-domain subgroup hiding and constant-size group signatures. In Tatsuaki Okamoto and Xiaoyun Wang, editors, *Public Key Cryptography*, volume 4450 of *Lecture Notes in Computer Science*, pages 1–15. Springer, 2007.

[CGH04] Ran Canetti, Oded Goldreich, and Shai Halevi. The random oracle methodology, revisited. *J. ACM*, 51(4):557–594, 2004.

[CvH91] David Chaum and Eugène van Heyst. Group signatures. In *EUROCRYPT*, pages 257–265, 1991.

[CZCG09] Wei Chen, Li Zhu, Xiaomei Cao, and Yang Geng. A novel fuzzy identity-based signature with dynamic threshold. In *NSS '09: Proceedings of the 2009 Third*

International Conference on Network and System Security, pages 192–198, Washington, DC, USA, 2009. IEEE Computer Society.

[Dam98] Ivan Damgård. Commitment schemes and zero-knowledge protocols. In Ivan Damgård, editor, *Lectures on Data Security*, volume 1561 of *Lecture Notes in Computer Science*, pages 63–86. Springer, 1998.

[DH76] Whitfield Diffie and Martin E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, IT-22(6):644–654, 1976.

[Fuc09] Georg Fuchsbauer. Automorphic signatures in bilinear groups and an application to round-optimal blind signatures. Cryptology ePrint Archive, Report 2009/320, 2009.

[Gam85] Taher El Gamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, 31(4):469–472, 1985.

[GHK06] David Galindo, Javier Herranz, and Eike Kiltz. On the generic construction of identity-based signatures with additional properties. In Xuejia Lai and Kefei Chen, editors, *ASIACRYPT*, volume 4284 of *Lecture Notes in Computer Science*, pages 178–193. Springer, 2006.

[GMR88] Shafi Goldwasser, Silvio Micali, and Ronald L. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM J. Comput.*, 17(2):281–308, 1988.

[Gol00] Oded Goldreich. *Foundations of Cryptography: Basic Tools*. Cambridge University Press, New York, NY, USA, 2000.

[GOS06] Jens Groth, Rafail Ostrovsky, and Amit Sahai. Perfect non-interactive zero knowledge for np. In Serge Vaudenay, editor, *EUROCRYPT*, volume 4004 of *Lecture Notes in Computer Science*, pages 339–358. Springer, 2006.

[GPS08] Steven D. Galbraith, Kenneth G. Paterson, and Nigel P. Smart. Pairings for cryptographers. *Discrete Applied Mathematics*, 156(16):3113–3121, 2008.

[Gro06] Jens Groth. Simulation-sound nizk proofs for a practical language and constant size group signatures. In Xuejia Lai and Kefei Chen, editors, *ASIACRYPT*, volume 4284 of *Lecture Notes in Computer Science*, pages 444–459. Springer, 2006.

[GS07] Jens Groth and Amit Sahai. Efficient non-interactive proof systems for bilinear groups. *Electronic Colloquium on Computational Complexity (ECCC)*, 14(053), 2007.

[GSW10] Essam Ghadafi, Nigel P. Smart, and Bogdan Warinschi. Groth-sahai proofs revisited. In Phong Q. Nguyen and David Pointcheval, editors, *Public Key Cryptography*, volume 6056 of *Lecture Notes in Computer Science*, pages 177–192. Springer, 2010.

[HLR10] Javier Herranz, Fabien Laguillaumie, and Carla Ràfols. Constant size ciphertexts in threshold attribute-based encryption. In Phong Q. Nguyen and David Pointcheval, editors, *Public Key Cryptography*, volume 6056 of *Lecture Notes in Computer Science*, pages 19–34. Springer, 2010.

[Jou02] Antoine Joux. The weil and tate pairings as building blocks for public key cryptosystems. In Claus Fieker and David R. Kohel, editors, *ANTS*, volume 2369 of *Lecture Notes in Computer Science*, pages 20–32. Springer, 2002.

[LAS+10] Jin Li, Man Ho Au, Willy Susilo, Dongqing Xie, and Kui Ren. Attribute-based signature and its applications. In Dengguo Feng, David A. Basin, and Peng Liu, editors, *ASIACCS*, pages 60–69. ACM, 2010.

[Men05] Alfred Menezes. An introduction to pairing-based cryptography. notes from lectures given in, 2005.

[QQQ+89] Jean-Jacques Quisquater, Myriam Quisquater, Muriel Quisquater, Michaël Quisquater, Louis C. Guillou, Marie Annick Guillou, Gaïd Guillou, Anna Guillou, Gwenolé Guillou, Soazig Guillou, and Thomas A. Berson. How to explain zero-knowledge protocols to your children. In Gilles Brassard, editor, *CRYPTO*, volume 435 of *Lecture Notes in Computer Science*, pages 628–631. Springer, 1989.

[RST01] Ronald L. Rivest, Adi Shamir, and Yael Tauman. How to leak a secret. In Colin Boyd, editor, *ASIACRYPT*, volume 2248 of *Lecture Notes in Computer Science*, pages 552–565. Springer, 2001.

[Sha84] Adi Shamir. Identity-based cryptosystems and signature schemes. In *CRYPTO*, pages 47–53, 1984.

[SSN09] Siamak Fayyaz Shahandashti and Reihaneh Safavi-Naini. Threshold attribute-based signatures and their application to anonymous credential systems. In Bart Preneel, editor, *AFRICACRYPT*, volume 5580 of *Lecture Notes in Computer Science*, pages 198–216. Springer, 2009.

[SW05] Amit Sahai and Brent Waters. Fuzzy identity-based encryption. In Ronald Cramer, editor, *EUROCRYPT*, volume 3494 of *Lecture Notes in Computer Science*, pages 457–473. Springer, 2005.

[SW07] Hovav Shacham and Brent Waters. Efficient ring signatures without random oracles. In Tatsuaki Okamoto and Xiaoyun Wang, editors, *Public Key Cryptography*, volume 4450 of *Lecture Notes in Computer Science*, pages 166–180. Springer, 2007.