HAL
archives-ouvertes.fr

Réputation et respect de la vie privée dans les réseaux dynamiques auto-organisés

Paul Lajoie-Mazenc

▶ **To cite this version:**

Paul Lajoie-Mazenc. Réputation et respect de la vie privée dans les réseaux dynamiques auto-organisés. Cryptographie et sécurité [cs.CR]. Université Rennes 1, 2015. Français. <NNT : 2015REN1S039>. <tel-01232139>

HAL Id: tel-01232139

https://tel.archives-ouvertes.fr/tel-01232139

Submitted on 23 Nov 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

ANNÉE 2015

THÈSE / UNIVERSITÉ DE RENNES 1
sous le sceau de l'Université Européenne de Bretagne

pour le grade de

DOCTEUR DE L'UNIVERSITÉ DE RENNES 1

Mention : Informatique

Ecole doctorale MATISSE

présentée par

Paul Lajoie-Mazenc

préparée à l'unité de recherche UMR 6074 IRISA
Institut de recherche en informatique et systèmes aléatoires
Université de Rennes 1

Réputation et respect de la vie privée dans les réseaux dynamiques et auto-organisés

Thèse soutenue à Rennes
le 25 septembre 2015

devant le jury composé de :

Sébastien CANARD
Ingénieur de recherche, Orange Labs / rapporteur

Maria POTOP-BUTUCARU
Professeur, Université Pierre et Marie Curie / rapporteur

Daniel LE MÉTAYER
Directeur de recherche, Inria / examinateur

Gildas AVOINE
Professeur, INSA Rennes / examinateur

Jean-Marc ROBERT
Professeur, École de Technologie Supérieure / président

Thomas SIRVENT
Ingénieur de recherche, DGA MI / examinateur

Emmanuelle ANCEAUME
Chargée de recherche, CNRS / directeur de thèse

Valérie VIET TRIEM TONG
Professeur assistant, CentraleSupélec / co-directeur de thèse

Résumé

Les mécanismes de réputation sont des outils très utiles pour inciter des utilisateurs ne se connaissant pas à se faire confiance, en récompensant les bons comportements et, inversement, en pénalisant les mauvais. Cependant, pour que la réputation des fournisseurs de service soit précise et robuste aux attaques, les mécanismes de réputation existants requièrent de nombreuses informations qui menacent la vie privée des utilisateurs ; par exemple, il est parfois possible de traquer les interactions effectuées par les clients. Des mécanismes de réputation préservant aussi bien la vie privée des clients que celle des fournisseurs sont donc apparus pour empêcher de telles attaques. Néanmoins, pour garantir des propriétés fortes de vie privée, ces mécanismes ont dû proposer des scores de réputation imprécis, notamment en ne permettant pas aux clients de témoigner de leurs interactions négatives.

Dans cette thèse, nous proposons un nouveau mécanisme de réputation distribué préservant la vie privée, tout en permettant aux clients d'émettre des témoignages négatifs. Une telle construction est possible grâce à des outils issus des systèmes distribués – des tierces parties distribuées qui permettent de distribuer la confiance et de tolérer des comportements malveillants – et de la cryptographie – par exemple des preuves de connaissance à divulgation nulle de connaissance ou des signatures proxy anonymes. Nous prouvons de plus que ce mécanisme garantit les propriétés de vie privée et de sécurité nécessaires, et montrons par des analyses théoriques et pratiques que ce mécanisme est utilisable.

Abstract

Reputation mechanisms are very powerful mechanisms to foster trust between unknown users, by rewarding good behaviors and punishing bad ones. Reputation mechanisms must guarantee that the computed reputation scores are precise and robust against attacks; to guarantee such properties, existing mechanisms require information that jeopardize users' privacy: for instance, clients' interactions might be tracked. Privacy-preserving reputation mechanisms have thus been proposed, protecting both clients' privacy and the providers' one. However, to guarantee strong privacy properties, these mechanisms provide imprecise reputation scores, particularly by preventing clients to testify about their negative interactions.

In this thesis, we propose a new distributed privacy-preserving reputation mechanism allowing clients to issue positive as well as negative feedback. Such a construction is made possible thanks to tools from the distributed systems community – distributed third parties that allow for a distribution of trust and that tolerate malicious behaviors – as well as from the cryptographic one – for instance zero-knowledge proofs of knowledge or anonymous proxy signatures. Furthermore, we prove that our mechanism guarantees the required privacy and security properties, and we show with theoretical and practical analysis that this mechanism is usable.

Remerciements

Je souhaite remercier les membres du jury, à commencer par Jean-Marc Robert qui l'a présidé. Sébastien Canard et Maria Potop-Butucaru ont accepté de rapporter cette thèse, tandis que Gildas Avoine et Thomas Sirvent ont accepté d'en être les examinateurs. Les apports de chacun d'entre eux ont permis d'améliorer significativement le manuscrit final.

Un grand merci à tous ceux qui m'ont encadré, et m'ont permis d'aboutir à ce manuscrit : mes directrices et encadrantes officielles, Emmanuelle Anceaume et Valérie Viet Triem Tong, et mes encadrants officieux, Gilles Guette et Thomas Sirvent. Vous m'avez tous énormément apporté : découverte du monde de la recherche, clarification de mes idées – et il y avait du travail ! –, approfondissement technique, et sur tant d'autres aspects. Je garderai en mémoire les sessions de dessin au tableau et les batailles de *commits* pendant les rédactions d'article !

Je suis arrivé à Rennes il y a un peu plus de quatre ans, et c'est l'équipe CIDRE qui m'y a accueilli – d'abord en tant qu'étudiant Supélec, puis en tant que doctorant. Je remercie tous ses membres pour l'ambiance formidable qui y règne ainsi que pour les discussions – très variées – des pauses café. Merci également à Lydie, qui parvient à gérer cette équipe et toutes ses particularités. Un remerciement particulier va à tous les non-permanents de l'équipe que j'ai côtoyé pendant ces années ; je risquerais d'en oublier si j'essayais de tous les énumérer. Vous contribuez tous à l'atmosphère particulièrement bonne de CIDRE ; continuez comme ça !

Finalement, je souhaite exprimer toute ma gratitude à ma famille et à mes proches pour leurs encouragements et leur profond soutien pendant ces années.

Table des matières

Liste des figures

Liste des tableaux

Introduction

Les communautés qui se sont développées sur Internet permettent à leurs membres d'interagir en s'échangeant des services, en se conseillant ou, dans le cas du commerce électronique, en se vendant des biens. Cependant, de telles communautés sont vastes, et un utilisateur ne connaît qu'une fraction négligeable des membres de ses communautés. Un problème se pose lorsqu'un client désire interagir avec un fournisseur de service qu'il ne connaît pas : tous deux s'investissent dans l'interaction, mais aucun des deux ne sait si son partenaire se comportera correctement, ce qui n'est pas le cas dans le monde physique : un magasin ayant pignon sur rue inspire a priori plus confiance qu'un inconnu sur Internet. Un risque non-négligeable est donc attaché à chaque interaction. Les mécanismes de réputation sont apparus comme des outils efficaces pour estimer ce risque [Res+00] : en effet, ils permettent aux clients de témoigner en notant leurs fournisseurs et d'utiliser ces témoignages pour calculer des scores de réputation, qui représentent les comportements passés des fournisseurs. Ainsi, un fournisseur se comportant incorrectement recevra des mauvaises notes, ce qui réduira sa réputation et donc son attractivité. Les mécanismes de réputation incitent donc les fournisseurs de service à se comporter correctement.

Certains mécanismes de réputation reposent sur une architecture centralisée pour collecter les notes et calculer les scores de réputation des utilisateurs [And+08 ; Ker09 ; SPT11]. En plus de rendre un passage à l'échelle problématique, une telle conception induit un point unique de défaillance : souvent, toute la sécurité du mécanisme de réputation repose sur l'autorité centrale. Les mécanismes distribués, bien que généralement plus complexes, permettent de tolérer des comportements byzantins [BSS10 ; Has10].

La collecte des témoignages et le calcul des scores de réputation doivent être robustes face aux tentatives de manipulation pour garantir que les scores de réputation représentent précisément le comportement des fournisseurs. À cet effet, certains mécanismes requièrent des informations sur les utilisateurs qui peuvent être considérées comme personnelles [Liu+11 ; WJI04]. Les témoignages peuvent de plus être considérés comme des données personnelles [Byg02] et doivent donc être protégés pour respecter la législation en vigueur [Eur95]. C'est pourquoi il est primordial de s'intéresser aux problèmes de vie privée dans les mécanismes de réputation. De nombreux mécanismes intègrent la protection de la vie privée de leurs utilisateurs au cœur de leur conception. Les premiers mécanismes proposés assurent le secret des témoignages [Has10 ; Ker09 ; PRT04] ; ils révèlent néanmoins l'identité des participants, ce qui est problématique. D'autres mécanismes, plus évolués, garantissent des propriétés plus fortes comme le k-anonymat des utilisateurs [CSK13] ou l'indistinguabilité des utilisateurs et l'inassociabilité de leurs interactions [And+08 ; BSS10].

Malheureusement, l'intégration de propriétés de vie privée fortes dans les mécanismes de réputation a conduit à une dégradation de l'utilité des scores de réputation : les mécanismes préservant correctement la vie privée ne permettent pas aux clients de laisser une trace de

leurs expériences négatives ; ces expériences sont pourtant indispensables pour nous faire une idée précise du comportement d'un fournisseur [Bau+01]. De plus, l'architecture même de ces mécanismes permet aux fournisseurs de n'utiliser qu'un sous-ensemble des témoignages reçus pour le calcul de leur score de réputation ; y ajouter des témoignages négatifs n'est donc pas trivial. Jusqu'à présent, la conception d'un mécanisme de réputation permettant aux clients de témoigner de leurs expériences négatives tout en préservant leur vie privée restait un problème ouvert. Comme indiqué par Bethencourt et coll., cette tâche n'est pas aisée :

> Most importantly, how can we support non-monotonic reputation systems, which can express and enforce bad reputation as well as good? Answering this question will require innovative definitions as well as cryptographic constructions. (Bethencourt et coll. [BSS10])

L'objectif de cette thèse est de concevoir un tel mécanisme.

Contributions

Cette thèse apporte plusieurs contributions aux mécanismes de réputation préservant la vie privée

Propriétés formelles de vie privée et de sécurité Avant de comparer des mécanismes de réputation préservant la vie privée, il est nécessaire de définir les propriétés à partir desquelles effectuer la comparaison. Nous proposons un ensemble de propriétés de vie privée et de sécurité utilisées pour comparer les mécanismes de l'état de l'art. Ces propriétés effectuent un compromis entre vie privée et sécurité, en garantissant un respect suffisamment fort de la vie privée des utilisateurs et en fournissant les informations nécessaires à un calcul précis des scores de réputation.

Mécanisme préservant la vie privée des clients Le premier mécanisme proposé s'intéresse à la vie privée des clients, sans considérer celle des fournisseurs. Il propose à cet effet une gestion distribuée des témoignages et des scores de réputation des fournisseurs pour garantir leur intégrité, et permet aux clients d'utiliser des pseudonymes pour interagir ; deux pseudonymes différents ne sont pas associables, ce qui garantit qu'un adversaire ne peut pas traquer les interactions d'un client. Ce mécanisme décrit également une nouvelle fonction de calcul de score de réputation qui ne nécessite pas d'informations personnelles ; des analyses évaluent la précision de cette fonction en présence d'utilisateurs malveillants.

Nouveaux outils cryptographiques et distribués La conception d'un mécanisme de réputation préservant la vie privée des clients et des fournisseurs nécessite des nouveaux outils. Nous présentons de nouveaux outils cryptographiques, dont l'*invariant*, ainsi que de nouvelles architectures distribuées pour la gestion des scores de réputations et des témoignages.

Mécanisme préservant la vie privée des utilisateurs À l'aide des outils introduits précédemment, nous présentons un mécanisme de réputation préservant la vie privée des utilisateurs, c'est-à-dire des clients et des fournisseurs de service. Nous en proposons également une variante, qui allège le coût imposé aux tierces parties distribuées. Nous

prouvons formellement que ce mécanisme garantit les propriétés de vie privée et de sécurité proposées.

Analyse des performances Finalement, nous analysons les performances du mécanisme de réputation préservant la vie privée des utilisateurs sur deux aspects : la taille des messages échangés par les participants d'une interaction, ainsi que leurs temps de calcul. Nous commençons par une analyse théorique, puis présentons les performances d'une implémentation en Python de ce mécanisme ; dans les deux cas, notre mécanisme est bien plus efficace que les mécanismes précédents.

Plan du manuscrit

Ce manuscrit est organisé en sept parties. Dans le chapitre 1, nous présentons le contexte général considéré. C'est-à-dire que nous précisons le vocabulaire utilisé pour parler des mécanismes de réputation, nous décrivons le modèle du système et de l'adversaire. Finalement, nous justifions et détaillons les propriétés de vie privée et de sécurité que nous utilisons. Avant de décrire l'état de l'art, le chapitre 2 présente quelques outils cryptographiques existants, accompagnés d'une introduction à l'algèbre et aux hypothèses cryptographiques classiques ; ce chapitre précise les définitions de sécurité utilisées par chacun des outils présentés. Ce chapitre nous permet d'arriver à l'état de l'art (chapitre 3), où nous présentons dans un premier temps les liens entre réputation et confiance et les propriétés nécessaires à la réputation ; nous décrivons ensuite les principales fonctions de calcul de réputation, les attaques sur les mécanismes de réputation et l'utilité de la vie privée pour la réputation. Nous finissons ce chapitre en exposant les principaux mécanismes de réputation préservant la vie privée et leurs limitations. Le chapitre 4 propose un premier mécanisme de réputation préservant la vie privée des clients, ainsi qu'une nouvelle fonction de calcul de score, qui y est évaluée. Le chapitre 5 redéfinit ce mécanisme plus formellement, puis introduit les outils nécessaires à un mécanisme de réputation préservant la vie privée des utilisateurs ; il détaille ensuite deux variantes d'un tel mécanisme. Le chapitre 6 prouve que ce mécanisme respecte les propriétés de vie privée et de sécurité présentées au chapitre 1. Finalement, le chapitre 7 valide notre mécanisme en montrant qu'il est efficace, aussi bien en théorie qu'en pratique.

1 Contexte général

Dans ce chapitre, nous commençons par présenter le contexte général de cette thèse. Dans un premier temps, nous définissons plus précisément les termes utilisés pour décrire les mécanismes de réputation. Une fois que cette terminologie est définie, nous précisons le modèle du système, ainsi que le modèle de l'adversaire. Finalement, nous sommes capable de présenter les propriétés de vie privée et de sécurité devant être garanties pour qu'un mécanisme de réputation soit considéré robuste et préservant la vie privée.

1.1 Mécanismes de réputation

Un mécanisme de réputation fait interagir au moins deux types d'utilisateurs : les *clients* et les *fournisseurs de service*. Les fournisseurs proposent des services que les clients désirent obtenir. Par exemple, ils vendent des biens sur une plateforme de commerce électronique, ils proposent des fichiers en téléchargement ou ils relisent des articles scientifiques ; nous ne spécifions pas la nature des échanges, qui peut être quelconque. Un même utilisateur peut être à la fois client et fournisseur dans différentes interactions.

Définition 1 (Client)
Un client *est un utilisateur désirant obtenir un service proposé par un fournisseur.*

Définition 2 (Fournisseur de service)
Un fournisseur *est un utilisateur proposant un ou plusieurs services, de nature quelconque.*

Une fois l'échange entre un client et un fournisseur terminé, le client a la possibilité de témoigner du comportement du fournisseur en déposant une *note* ; si le client est satisfait, la note est dite *positive*, et *négative* sinon.

Définition 3 (Note)
Une note *représente l'opinion d'un client sur un fournisseur. Lorsque le client est satisfait, la note est dite* positive. *Sinon, la note est dite* négative.

Tous les mécanismes de réputation ne permettent pas aux clients d'émettre des notes négatives [And+08 ; BSS10]. Nous expliquons en section 3.1 pourquoi celles-ci sont nécessaires. Un client peut décider de ne pas laisser de note ; c'est en général le cas des clients satisfaits. Ainsi, la majorité des mauvais comportements des fournisseurs sont pris en compte, mais seulement une fraction de leurs bons comportements le sont. Afin de répondre à cette asymétrie, nous proposons aux fournisseurs de service d'obtenir des *preuves de transaction* qui témoignent des transactions, et permettent de distinguer deux fournisseurs ayant reçu les mêmes notes : le fournisseur ayant obtenu le plus de preuves de transaction est probablement le plus fiable.

Définition 4 (Preuve de transaction)
Une preuve de transaction *atteste qu'un fournisseur de service a participé à une transaction, même lorsque le client n'a pas témoigné.*

La preuve de transaction, accompagnée ou non de la note, dresse le bilan de la transaction : le *témoignage*. Certains mécanismes de réputation incorporent en plus d'autres données dans le témoignage, comme la date de la transaction ou sa valeur [JI02].

Définition 5 (Témoignage)
Un témoignage *est le compte-rendu d'une transaction, qui contient la preuve de transaction, et éventuellement la note du client et d'autres données.*

Finalement, les témoignages des clients passés d'un fournisseur – ses *témoins* – sont combinés pour obtenir son *score de réputation*, qui résume son comportement passé.

Définition 6 (Témoin)
Les témoins *d'un fournisseur de service sont ses clients passés.*

Définition 7 (Score de réputation)
Le score de réputation *d'un fournisseur de service est un résumé de son comportement passé.*

Le score de réputation – ou, plus simplement, la réputation – d'un fournisseur croît lorsqu'il reçoit de nouveaux témoignages positifs, et décroît lorsque ceux-ci sont négatifs. Le format des témoignages et des scores de réputation, ainsi que la manière de les calculer sont décrits par le *moteur de réputation*.

Définition 8 (Moteur de réputation)
Le moteur de réputation *spécifie le format des témoignages et des scores de réputation, ainsi que la manière de combiner les témoignages pour obtenir un score de réputation.*

Nous en donnons plusieurs exemples en section 3.2.

Tout au long de cette thèse, nous différencions les *transactions* des *interactions*. Une transaction représente un échange de service entre un fournisseur et un client, tandis qu'une interaction correspond à l'exécution du protocole suivi par le client et le fournisseur, permettant au client de vérifier la réputation du fournisseur, d'obtenir le service, et finalement d'émettre un témoignage comprenant sa note.

Définition 9 (Transaction)
Une transaction *correspond à l'échange d'un service entre un client et un fournisseur.*

Définition 10 (Interaction)
Une interaction *comprend l'ensemble des communications permettant à un client et un fournisseur de service d'effectuer une transaction, de la vérification de la réputation du fournisseur à l'émission du témoignage. Plus précisément, une interaction se déroule en quatre étapes :*

1. *la vérification de la réputation du fournisseur ;*
2. *la préparation à la transaction, qui est optionnelle ;*

3. *la transaction ;*
4. *l'émission du témoignage, qui permet au client de choisir une note.*

Les utilisateurs peuvent décider de se comporter conformément au protocole défini par le mécanisme de réputation ; par exemple, les clients sont supposés noter les fournisseurs à la fin de leurs interactions, tandis que les fournisseurs sont sensés les y aider. Nous disons qu'un tel utilisateur est *honnête*, tandis qu'un utilisateur dont le comportement dévie du protocole est *malveillant*. Notons qu'un utilisateur honnête peut être *curieux*, c'est-à-dire qu'il peut chercher à apprendre les informations secrètes des autres utilisateurs, tout en suivant le protocole.

Définition 11 (Honnêteté)
Un utilisateur est dit honnête *s'il se conforme au protocole décrit par le mécanisme de réputation. Autrement, il est dit* malveillant.

Définition 12 (Curiosité)
Un utilisateur curieux *est un utilisateur honnête qui cherche à apprendre les informations secrètes des autres utilisateurs à partir des interactions légitimes entre utilisateurs honnêtes.*

Similairement, nous disons qu'un témoignage ou un score de réputation est *valide* s'il est accepté par un utilisateur honnête, et *invalide* dans le cas contraire. La définition exacte de validité d'un témoignage dépend du mécanisme de réputation, mais est essentielle pour garantir l'infogeabilité des témoignages et des preuves de réputation (voir leur définition en section 1.4). Cette définition, bien que manquant de précision, permet d'être compatible avec tout mécanisme de réputation, sans perte de généralité.

Définition 13 (Validité d'un témoignage ou d'une réputation)
Un témoignage ou un score de réputation est valide *si et seulement si il est accepté par un utilisateur honnête. Sinon, il est* invalide.

Nous définissons plus formellement la validité d'un témoignage ou d'un score de réputation dans le contexte du mécanisme de réputation proposé lors des preuves des propriétés de sécurité (voir sections 6.1.4 et 6.1.5).

1.2 Modèle du système

Nous considérons un système ouvert, permettant à n'importe quel utilisateur de le rejoindre ou de le quitter. Ainsi, le système est potentiellement à large échelle, c'est pourquoi les algorithmes utilisés doivent être efficaces. Avant qu'un utilisateur n'entre dans le système, certains mécanismes de réputation imposent qu'il s'enregistre auprès d'une *autorité centrale*, par exemple pour obtenir des identifiants ou des clés cryptographiques [And+08 ; BSS10 ; Has10 ; Ker09]. Cet enregistrement peut requérir un coût, monétaire ou calculatoire, pour éviter qu'un seul utilisateur ne se crée trop de comptes et ne lance une attaque Sybil [Bor06 ; Dou02] ; nous donnons plus d'informations sur de telles attaques en section 3.3. Une fois enregistré, un utilisateur peut agir à la fois en tant que client et en tant que fournisseur de service. Finalement, nous supposons que les utilisateurs communiquent à travers un réseau de communication anonyme pour empêcher la surveillance de leurs adresses IP ; à cet effet, ils peuvent utiliser un réseau de routage en oignon [SGR97] comme Tor [DMS04].

Hypothèse 1 (Communications anonymes)
Les utilisateurs communiquent à travers un réseau de communication anonyme.

1.3 Modèle de l'adversaire

Les mécanismes de réputation considèrent généralement un – ou plusieurs – modèles d'adversaire parmi les trois suivants :

Adversaire curieux Un utilisateur curieux suit le protocole spécifié par le mécanisme, mais essaye d'apprendre les secrets des autres utilisateurs grâce aux messages reçus et aux calculs effectués.

Adversaire malveillant non-disruptif À l'instar des utilisateurs curieux, un utilisateur malveillant non-disruptif a pour objectif d'apprendre les secrets des autres utilisateurs. Afin d'atteindre cet objectif, un tel utilisateur peut adopter n'importe quel comportement ; il peut par exemple refuser de participer à une interaction, émettre des notes aléatoires, ou empêcher le routage correct des messages. Un adversaire non-disruptif se comporte de manière malveillante uniquement pour atteindre son objectif.

Adversaire malveillant disruptif En plus de vouloir apprendre les secrets des autres utilisateurs, un utilisateur malveillant disruptif – ou byzantin – peut également vouloir empêcher les utilisateurs honnêtes de participer au protocole ; à cet effet, il peut se comporter de manière arbitraire.

Un utilisateur curieux est moins puissant qu'un utilisateur malveillant non-disruptif, qui est lui-même moins puissant qu'un utilisateur malveillant disruptif. Dans la suite, nous considérons que tout utilisateur, client ou fournisseur de service, peut être malveillant disruptif.

Hypothèse 2 (Comportement des utilisateurs)
Tout utilisateur peut être malveillant disruptif.

L'adversaire peut être *adaptatif* ou *non-adaptatif*. Dans le premier cas, l'adversaire peut corrompre des utilisateurs au fur et à mesure du déroulement du protocole, tandis que dans le second, les utilisateurs malveillants sont fixés au début du protocole. Nous détaillons les hypothèses sur le modèle de l'adversaire lors de la présentation des mécanismes de réputation.

Finalement, nous supposons que l'adversaire a une puissance de calcul limitée. Plus précisément, nous le modélisons par une machine de Turing probabiliste en temps polynomial (PPT). C'est-à-dire que l'adversaire peut exécuter des algorithmes probabilistes dont le temps d'exécution est polynomial en la longueur des entrées. Notamment, l'adversaire ne peut pas tester toutes les combinaisons d'un mot de passe pour le deviner : il y a c^ℓ mots de passe possibles de longueur ℓ sur un alphabet de c caractères, ce qui est exponentiel et non polynomial en ℓ. Cette modélisation est courante en cryptographie ; Katz et Lindell introduisent cette approche [KL07, § 3.1].

Hypothèse 3 (Puissance de calcul de l'adversaire)
L'adversaire est une machine de Turing PPT.

1.4 Propriétés de vie privée et de sécurité

Dans cette section, nous formulons et motivons les propriétés de vie privée et de sécurité qu'un mécanisme de réputation doit garantir afin d'être sécurisé et de préserver la vie privée de ses utilisateurs. Notons que nous ne présentons ici que des vues intuitives de ces propriétés ; leur énoncé formel est présenté en même temps que leurs preuves, au Chapitre 6.

1.4.1 Propriétés de vie privée

Nous expliquons en section 3.4 pourquoi les mécanismes de réputation doivent préserver la vie privée de leurs utilisateurs. Les mécanismes utilisés couramment – par exemple le site d'enchères électroniques eBay [1] – permettent à leurs utilisateurs de choisir un pseudonyme. Cependant, ce pseudonyme est un identifiant unique auquel toutes les interactions et témoignages de cet utilisateur sont rattachés, ce qui permet d'obtenir des informations. Préserver la vie privée des utilisateurs n'est donc pas chose aisée, et les termes de cette préservation doivent être définis avec précaution. Dans le contexte des mécanismes de réputation, nous considérons les propriétés de vie privée suivantes :

Propriété 1 (Vie privée du fournisseur de service). *Au moment où un client note un fournisseur de service honnête, ce fournisseur est anonyme parmi tous les fournisseurs de service honnêtes ayant la même réputation.*

Propriété 2 (Vie privée du client). *Au moment où un fournisseur de service procède à la transaction avec un client honnête, ce client est indistinguable de tout autre client honnête. De plus, les interactions des clients honnêtes avec différents fournisseurs de service ne sont pas associables.*

Préserver la vie privée des fournisseurs vise à empêcher les clients de choisir un fournisseur sur d'autres critères que le service fourni ou sa réputation ; ainsi, les clients ne peuvent sélectionner un fournisseur particulier pour diminuer sa réputation. De plus, les client ne doivent choisir leur note qu'en fonction du comportement du fournisseur pendant la transaction. Il faut donc que les fournisseurs de service soient anonymes au moment où les clients choisissent leur note ; c'est-à-dire, suivant la définition de Pfitzmann et Hansen [PH10], que les clients ne doivent pas être capables de distinguer un fournisseur parmi les fournisseurs ayant une même réputation. Ainsi, l'identité du fournisseur de service ne peut pas avoir d'influence sur la note choisie par le client.

Remarque
Notons que le client peut distinguer un fournisseur de haute réputation d'un fournisseur de basse réputation : le but des mécanismes de réputation est de permettre cette distinction. De plus, cette propriété ne mentionne pas le moteur de réputation employé, c'est-à-dire la manière dont sont calculés les scores de réputation. Notamment, un mécanisme de réputation assignant une réputation unique à chaque fournisseur de service peut masquer leurs identifiants ; néanmoins, le client connaît la réputation du fournisseur, et peut donc traquer son comportement. Concevoir un mécanisme de réputation préservant la vie privée requiert donc de choisir

1. https://www.ebay.com

un moteur de réputation adapté, qui garantit que l'ensemble d'anonymat des fournisseurs est suffisamment grand pour préserver la vie privée des fournisseurs ; le mécanisme proposé par Clauß et coll. [CSK13] garantit qu'au moins k fournisseurs possèdent la même réputation.

L'objet de cette thèse n'est pas de concevoir un moteur de réputation préservant la vie privée, mais de fournir tous les éléments nécessaires au calcul robuste des scores de réputation, c'est-à-dire de garantir les propriétés de sécurité décrites ci-dessous. La plupart des moteurs existants – présentés au chapitre 3 – peuvent être adaptés pour garantir que les ensembles d'anonymat des fournisseurs sont suffisamment grands. Dans la suite, nous supposons que le moteur de réputation utilisé est adapté. ■

Préserver la vie privée des clients empêche un fournisseur de traquer un client donné, et par exemple de le discriminer en lui offrant systématiquement un service de moindre qualité. Ainsi, il faut éviter qu'un fournisseur ne sache avec quel client il interagit lorsque la transaction se déroule. De plus, si plusieurs fournisseurs combinent les informations à propos de leurs témoins, ils peuvent potentiellement reconstruire le profil de leurs clients en effectuant des attaques par inférence [BDK07 ; NS08] ; ces deux articles parviennent à désanonymiser des utilisateurs à partir de bribes d'information. Il faut donc veiller à minimiser la quantité d'information divulguée. C'est pour cela qu'il faut empêcher d'associer les clients de différents fournisseurs.

1.4.2 Propriétés de sécurité

Les propriétés de sécurité peuvent se séparer en plusieurs catégories. Tout d'abord, il faut garantir que ni le client ni le fournisseur ne peuvent empêcher l'émission du témoignage.

Propriété 3 (Indéniabilité des notes). *À la fin d'une transaction entre un client honnête et un fournisseur, le fournisseur ne peut pas empêcher un client de le noter et de construire un témoignage valide, qui sera pris en compte dans le score de réputation du fournisseur.*

Propriété 4 (Indéniabilité des preuves de transaction). *À la fin d'une transaction entre un client et un fournisseur honnête, le client ne peut pas empêcher le fournisseur d'obtenir un témoignage valide, comprenant la preuve de transaction.*

Si ces propriétés ne sont pas garanties, un fournisseur peut mal se comporter et empêcher le client de le noter, garantissant de cette façon que son score de réputation ne peut diminuer. Les notes négatives ne sont utiles que si l'indéniabilité des notes est garantie ; dans le cas contraire, un fournisseur peut empêcher les clients d'émettre des notes négatives tout en continuant à recevoir des notes positives. De la même façon, le client pourrait empêcher un fournisseur de recevoir une preuve de transaction.

Ensuite, il faut garantir qu'un utilisateur malveillant ne peut pas forger un témoignage ou un score de réputation.

Remarque
Deux utilisateurs malveillants peuvent créer un témoignage sans qu'une transaction n'ait eu lieu ; la forge d'un témoignage n'est donc valide que si elle concerne un utilisateur honnête. ■

Propriété 5 (Inforgeabilité des témoignages). *Si un témoignage concernant un client et un fournisseur de service est valide, et que le client ou le fournisseur est honnête, alors ce témoignage a été émis à l'issue d'une transaction entre ces deux utilisateurs.*

Propriété 6 (Inforgeabilité des scores de réputation). *Un fournisseur de service ne peut pas forger un score de réputation valide différent de celui calculé à partir de ses témoignages.*

Sans ces propriétés, un utilisateur malveillant serait capable soit d'augmenter sa réputation à volonté, soit de diminuer celle d'un concurrent.

La dernière propriété nécessaire concerne l'associabilité des témoignages.

Propriété 7 (Associabilité des témoignages). *Deux témoignages valides émis par le même client sur le même fournisseur sont publiquement associables.*

Cette propriété permet d'empêcher les *bourrages d'urne* [Del00], que nous détaillons en section 3.3.3. Le principe de cette attaque est le suivant : dans un mécanisme de réputation, un fournisseur de service peut interagir avec lui-même, et se donner des bonnes notes pour augmenter sa réputation. Si le mécanisme de réputation préserve la vie privée des clients, il est impossible de distinguer un tel témoignage du témoignage d'un client quelconque. Pire encore, le fournisseur peut procéder ainsi de nombreuses fois afin d'obtenir une réputation massivement positive. Pour empêcher de telles attaques, nous proposons d'*associer* les témoignages à propos d'un même fournisseur de service, c'est-à-dire d'être capable de dire si le même client a émis plusieurs témoignages sur un même fournisseur de service. Notons que cette propriété n'est pas incompatible avec la vie privée des clients (Propriété 2), qui s'intéresse à l'associabilité entre *différents* fournisseurs de service et non pas à l'associabilité des clients d'un même fournisseur.

1.5 Résumé

Dans ce chapitre, nous avons commencé par préciser les termes employés pour décrire à la fois un mécanisme de réputation et le comportement des utilisateurs. Nous avons également détaillé le modèle du système et trois modèles réalistes de l'adversaire. Finalement, nous avons présenté et justifié les sept propriétés qu'un mécanisme de réputation doit garantir pour être sécurisé et préserver la vie privée de ses utilisateurs. Il est intéressant de noter qu'aucune des propriétés définies ne requiert un moteur de réputation spécifique.

2 Outils cryptographiques existants

L'objectif de ce chapitre est de présenter les outils cryptographiques existants avec lesquels nous construisons un mécanisme de réputation préservant la vie privée de ses utilisateurs tout en permettant aux clients d'émettre des témoignages positifs et négatifs. Avant de présenter ces outils, nous rappelons des notions d'algèbre des groupes et des groupes bilinéaires (section 2.1), ainsi que des bases de cryptographie (section 2.2).

Lors de la description d'un outil, nous commençons par décrire les propriétés de sécurité garanties par cet outil, puis nous détaillons son instanciation.

Nous présentons quatre outils cryptographiques : deux schémas de partage de secret (voir section 2.3) [PRT04 ; Sha79] ; un schéma d'engagement SXDH proposé par Groth et Sahai [GS08] (voir section 2.4) ; le système de preuve construit à partir de ces engagements [GS08] (voir section 2.5) ; trois schémas de signature : un schéma de signature classique [KW03], un schéma dit *automorphe* [Abe+10], qui permet d'instancier un schéma de signature *proxy anonyme* [FP08].

2.1 Rappels d'algèbre

Nous commençons par rappeler la notion de groupe, et introduisons les notations que nous utiliserons dans ce document. Nous expliquons ensuite ce que sont les couplages ainsi que les groupes bilinéaires, qui nous serviront à présenter la construction d'outils cryptographiques. Pour plus de détails, Shoup présente une introduction détaillée à la théorie des nombres et à l'algèbre à destination des informaticiens [Sho08].

2.1.1 Groupes

La notion de groupe est fondamentale en algèbre et en cryptographie moderne. Un groupe est défini comme suit :

Définition 14 (Groupe)
Soient $\mathbb{G}$ un ensemble et $\circ$ une opération binaire sur $\mathbb{G}$, c'est-à-dire une fonction prenant deux éléments de $\mathbb{G}$ en entrée. L'ensemble $\mathbb{G}$ muni de l'opération $\circ$ est un groupe si et seulement si les conditions suivantes sont vérifiées :

clôture *Pour tous $G, H \in \mathbb{G}$, $G \circ H \in \mathbb{G}$.*

existence d'un neutre *Il existe un élément $E \in \mathbb{G}$ tel que pour tout $G \in \mathbb{G}, E \circ G = G \circ E = G$; cet élément est appelé le* neutre *de $\mathbb{G}$.*

existence des inverses *Pour chaque élément $G \in \mathbb{G}$, il existe $H \in \mathbb{G}$ tel que $G \circ H = H \circ G = E$; H est appelé l'*inverse *de G et est noté G^{-1}.*

associativité *Pour tous $G_1, G_2, G_3 \in \mathbb{G}$, $(G_1 \circ G_2) \circ G_3 = G_1 \circ (G_2 \circ G_3)$.*

Remarques
Soit $(\mathbb{G}, \circ)$ un groupe de neutre E.

— Quand un groupe $\mathbb{G}$ possède un nombre fini d'éléments, ce groupe est dit *fini*, et le nombre d'éléments de ce groupe, noté $|\mathbb{G}|$, est appelé l'*ordre* du groupe.

— Si l'opération $\circ$ est commutative, c'est-à-dire que pour tous $G, H \in \mathbb{G}$, $G \circ H = H \circ G$, alors le groupe est dit *abélien*.

— Soit $G \in \mathbb{G}$. On dit que G est un *générateur* de $\mathbb{G}$ et que $\mathbb{G}$ est *engendré* par G si tous les éléments de $\mathbb{G}$ peuvent être atteints en combinant G et son inverse avec l'opération $\circ$; c'est-à-dire que $\mathbb{G} = \{\ldots, G^{-1}, E, G \circ G, \ldots\}$. ■

Exemples

— L'ensemble $\mathbb{Z}$ des entiers, muni de l'addition, est un groupe ; son neutre est 0, et l'inverse d'un nombre n est $-n$.

— L'ensemble $\mathbb{Z}_p = \{0, \ldots, p-1\}$ muni de l'addition modulo p est un groupe ; son neutre est 0, et l'inverse de n est $(-n \mod p)$.

— L'ensemble $\mathbb{Z}_p^* = \{n \in \mathbb{Z}_p \mid \text{pgcd}(n,p) = 1\}$ muni de la multiplication modulo p est également un groupe dont le neutre est 1 ; l'inverse d'un nombre peut être déterminé en utilisant l'algorithme d'Euclide étendu [Sho08, § 4.2]. En particulier, si p est premier, $\mathbb{Z}_p^* = \mathbb{Z}_p \backslash \{0\}$.

— Les courbes elliptiques sur des corps finis ont également une structure de groupe ; Silverman détaille leur arithmétique [Sil09] et Hankerson et coll. présentent leur utilité en cryptographie [HMV04]. ■

Dans la suite, nous utilisons les notations suivantes :

— les groupes sont notés multiplicativement : l'opération est notée $\cdot$ et nous écrivons $G^m = \underbrace{G \cdot \cdots \cdot G}_{m \text{ fois}}$, où m est un *scalaire*, c'est-à-dire un élément de $\mathbb{Z}_p$;

— les éléments de groupe sont désignés par des lettres capitales : $G, H \in \mathbb{G}$;

— les scalaires sont notés par des lettres minuscules : $r, s \in \mathbb{Z}_p$;

— les fonctions sont notées en caractères romans : Sign, Enc ;

— les variables désignant un élément de groupe ou un scalaire sont notées en police de caractères sans empattement : cert, nym ;

— si G est un multiplet, alors $G_{\mathbf{1}}$ désigne le premier élément de ce multiplet et, plus généralement, $G_{\mathbf{k}}$ en désigne le k-ième élément : l'indice désignant l'élément du multiplet considéré est en gras ;

— le signe « $\leftarrow$ » désigne le résultat d'une fonction probabiliste : $\text{crs} \leftarrow \text{Crs}(\kappa)$;

— le signe « $=$ » désigne le résultat d'une fonction déterministe : $Y = G^x$;

— le signe « $\xleftarrow{R}$ » désigne le choix uniformément aléatoire d'un élément dans un ensemble : $x \xleftarrow{R} \mathbb{Z}_p$ désigne un scalaire choisi aléatoirement.

2.1.2 Groupes bilinéaires et couplages

Définition 15 (Couplage)
Soient $(\mathbb{G}_1,\cdot)$, $(\mathbb{G}_2,\cdot)$ *et* $(\mathbb{G}_T,\cdot)$ *trois groupes d'ordre* p *premier. Soient* G_1 *et* G_2 *des générateurs respectifs de* $\mathbb{G}_1$ *et* $\mathbb{G}_2$*. Soit* e *une application :*

$$e : \begin{array}{ccc} \mathbb{G}_1 \times \mathbb{G}_2 & \longrightarrow & \mathbb{G}_T \\ A, B & \mapsto & e(A,B) \end{array}$$

L'application e *est un* couplage *si elle respecte les conditions suivantes :*

bilinéaire *pour tous* $A_1, A_2 \in \mathbb{G}_1$ *et* $B_1, B_2 \in \mathbb{G}_2$, $e(A_1 \cdot A_2, B_1) = e(A_1,B_1) \cdot e(A_2,B_1)$ *et* $e(A_1, B_1 \cdot B_2) = e(A_1,B_1) \cdot e(A_1,B_2)$ *;*

non dégénérée $e(G_1,G_2) \neq 1_T$*, où* 1_T *est le neutre de* $\mathbb{G}_T$ *;*

facilement calculable *un algorithme permet d'évaluer* e *efficacement.*

Les couplages sont très intéressants en cryptographie car ils permettent de construire des schémas de signature courte [BB08 ; BBS04], d'effectuer des échanges de clés tripartites en un seul tour [Jou00], de construire des schémas de chiffrement fondés sur l'identité [BF01] ou des systèmes de preuves de connaissance à divulgation nulle de connaissance efficaces [GS08]. Les couplages sont définis dans le cadre de groupes bilinéaires :

Définition 16 (Groupe bilinéaire)
Un groupe bilinéaire $\Lambda = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, G_1, G_2)$ *est composé de*

- *trois groupes* $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ *d'ordre* p *premier ;*
- *un couplage* $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ *;*
- G_1 *un générateur de* $\mathbb{G}_1$ *et* G_2 *un générateur de* $\mathbb{G}_2$*.*

En pratique, Galbraith et coll. classifient les groupes bilinéaires suivant trois types [GPS06] :

Type 1 Dans le premier type de groupe bilinéaire, $\mathbb{G}_1 = \mathbb{G}_2$ dans la définition du groupe bilinéaire.

Type 2 Dans le deuxième type de groupe bilinéaire, $\mathbb{G}_1 \neq \mathbb{G}_2$, mais il existe un homomorphisme $\phi : \mathbb{G}_2 \to \mathbb{G}_1$ calculable efficacement ; cet homomorphisme n'est pas un isomorphisme, c'est-à-dire qu'il n'est pas inversible : il est facile de passer de $\mathbb{G}_2$ à $\mathbb{G}_1$, mais l'inverse n'est pas vrai.

Type 3 Dans le dernier type de groupe bilinéaire, il n'existe pas d'homomorphisme calculable efficacement entre $\mathbb{G}_1$ et $\mathbb{G}_2$, dans un sens ou dans l'autre.

Plusieurs travaux introduisent la cryptographie utilisant les couplages [GPS08 ; Men09].

2.2 Hypothèses cryptographiques

Les schémas cryptographiques modernes reposent presque toujours sur l'hypothèse qu'un problème est *difficile* à résoudre. Par exemple, le schéma de chiffrement RSA [RSA78] repose sur l'hypothèse qu'il est difficile de factoriser un nombre. En pratique, difficile signifie qu'il

n'existe pas d'algorithme *polynomial* résolvant le problème. Par exemple, l'algorithme vérifiant de manière exhaustive si $p = 2, \ldots, \sqrt{n}$ divise n prend un temps $O(\sqrt{n} \cdot (\log n)^c)$ pour une constante c ; la longueur de n est $\log n$, cet algorithme est donc exponentiel en la longueur de n et n'est ainsi pas efficace.

Les hypothèses sont présentées sous la forme d'expériences – ou jeux – cryptographiques, impliquant un adversaire $\mathcal{A}$. Comme expliqué en section 1.3, cet adversaire est généralement modélisé par une machine de Turing PPT. Une des hypothèses couramment effectuée en cryptographie est celle du logarithme discret, qui est définie pour un algorithme de génération de groupe $\mathcal{G}$, un adversaire $\mathcal{A}$ et un paramètre de sécurité κ – ce paramètre représente le niveau de sécurité désiré, par exemple 80 bit, 128 bit ou plus – par l'expérience suivante : [1]

Logarithme discret $\mathrm{DLog}_{\mathcal{A},\mathcal{G}}(\kappa)$

1 : $(\mathbb{G}, p, G) \leftarrow \mathcal{G}(\kappa)$, où $\mathbb{G}$ est un groupe d'ordre p, un nombre de κ bit, engendré par G

2 : $H \xleftarrow{R} \mathbb{G}$

3 : $x \in \mathbb{Z}_p \leftarrow \mathcal{A}(\mathbb{G}, p, G, H)$, c'est-à-dire que $\mathcal{A}$ obtient $\mathbb{G}$, p, G, H, et renvoie $x \in \mathbb{Z}_p$

4 : L'expérience renvoie 1 si $G^x = H$, et 0 sinon.

La probabilité que ce jeu renvoie 1, prise sur tous les choix aléatoires de $\mathcal{A}$ et de $\mathcal{G}$, est appelée l'*avantage* de l'adversaire, qui est noté $\mathrm{Av}_{\mathrm{DLog}_{\mathcal{A},\mathcal{G}}}(\kappa)$. Un problème est dit *difficile* si l'avantage de l'adversaire pour le résoudre est « très faible », c'est-à-dire *négligeable*.

Définition 17
Soit f une fonction de $\mathbb{N}$ à valeurs réelles. f est négligeable *si, pour tout polynôme p,*

$$\exists n_0 \ t.q. \ \forall n > n_0, \quad f(n) < \frac{1}{p(n)}.$$

Définition 18
Le problème du logarithme discret est difficile *pour $\mathcal{G}$ si, pour tout adversaire PPT $\mathcal{A}$, il existe une fonction négligeable* neg *telle que*

$$\mathrm{Av}_{\mathrm{DLog}_{\mathcal{A},\mathcal{G}}}(\kappa) \leqslant \mathrm{neg}(\kappa).$$

L'hypothèse du logarithme discret est l'hypothèse qu'il existe un algorithme de génération de groupe $\mathcal{G}$ qui renvoie des groupes dans lesquels le problème du logarithme discret est difficile. Attention, ce problème n'est pas difficile dans tous les groupes ; par exemple, il existe un algorithme efficace permettant de le résoudre dans $(\mathbb{Z}_p, +)$: l'algorithme d'Euclide étendu permet de l'obtenir avec une complexité polynomiale de $O\left((\log p)^2\right)$.

D'autres problèmes reposant sur celui du logarithme discret ont également été proposés, comme les problèmes de la famille Diffie Hellman. Le premier de ces problèmes est le Diffie Hellman calculatoire (CDH). Informellement, ce problème est défini de la manière suivante. Soit $\mathbb{G}$ un groupe engendré par G. Pour deux éléments H_1, H_2 de $\mathbb{G}$, $\mathrm{DH}_G(H_1, H_2)$ est défini comme

$$\mathrm{DH}_G(H_1, H_2) = G^{\log_G H_1 \cdot \log_G H_2} = {H_1}^{\log_G H_2} = {H_2}^{\log_G H_1}.$$

1. Les définitions présentées dans cette section sont celles proposées par Katz et Lindell [KL07, § 7.3].

Le problème CDH consiste à calculer $\mathrm{DH}_G(H_1, H_2)$ en connaissant uniquement $\mathbb{G}$, G, H_1 et H_2. Notons que s'il est facile de résoudre le logarithme discret dans un groupe, alors il est également facile de résoudre le CDH en calculant $\log_G H_1$ ou $\log_G H_2$.

Le problème du Diffie Hellman décisionnel (DDH) est, informellement, de distinguer un triplet (G^x, G^y, G^{xy}) d'un triplet (G^x, G^y, G^z) où x, y et z sont choisis aléatoirement. Plus formellement, pour un algorithme de génération de groupe $\mathcal{G}$, ce problème est défini par :

Définition 19
Le problème DDH est difficile *pour un algorithme de génération de groupe $\mathcal{G}$ si, pour tout adversaire PPT $\mathcal{A}$, il existe une fonction négligeable* neg *telle que*

$$\mathrm{Av}_{DDH}(\kappa) = \left| P\left[\mathcal{A}(\mathbb{G}, p, G, G^x, G^y, G^z) = 1\right] - P\left[\mathcal{A}(\mathbb{G}, p, G, G^x, G^y, G^{xy}) = 1\right] \right| \leqslant \mathrm{neg}(\kappa)$$

Notons que si le problème CDH est facile à résoudre, alors il est également facile de résoudre le problème DDH en calculant G^{xy} à partir de G, G^x et G^y. Notons également que dans un groupe bilinéaire de type 1 (définition 16), il est facile de résoudre le problème DDH dans le groupe $\mathbb{G}_1 = \mathbb{G}_2$: en effet, il est possible de calculer $e(G^x, G^y)$ et $e(G, G^z)$. Grâce à la bilinéarité du couplage, ces deux valeurs sont égales si et seulement si $z = xy$. Dans un groupe bilinéaire de type 2, il est possible de résoudre ce problème dans le groupe $\mathbb{G}_2$: en effet, $e(\phi(G_2{}^x), G_2{}^y) = e(\phi(G_2), G_2{}^z)$ si et seulement si $z = xy$.

Dans le cadre des groupes bilinéaires, nous nous intéressons plus particulièrement à l'hypothèse Diffie Hellman symétrique externe (SXDH) [Bal+05]. Cette hypothèse est valide dans un groupe bilinéaire Λ si le problème DDH est difficile dans les groupes $\mathbb{G}_1$ et $\mathbb{G}_2$ de Λ. Notons que cette hypothèse n'est valide que dans des groupes bilinéaire de type 3.

2.3 Partage de secret

Le partage de secret [Sha79] est un schéma cryptographique à seuil, permettant de générer n parts à partir d'un secret. Le secret peut ensuite être reconstruit à partir d'un nombre prédéfini t de parts (avec $t \leqslant n$), sans donner aucune information sur le secret à quiconque possède strictement moins de t parts. Il existe plusieurs manières de partager un secret. Nous présentons ici deux méthodes ; la première, utilisée par exemple par Pavlov et coll. [PRT04], permet de partager un nombre avec un seuil $t = n$, tandis que la seconde est plus générale et permet de partager un élément de groupe avec un seuil quelconque [Sha79].

2.3.1 Partage d'un nombre

Le partage de secret utilisé par Pavlov et coll. [PRT04] ainsi que par Hasan [Has10] – deux mécanismes de réputation que nous détaillons en section 3.5.1 – permet de partager un nombre en n parts, et de le reconstruire en possédant les n parts.

Notons secret le nombre à partager. Pour partager le secret en n parts, il faut choisir aléatoirement $n-1$ nombres $s_1, \ldots, s_{n-1} \xleftarrow{R} \mathbb{R}$. La dernière part est $s_n = \text{secret} - \sum_{i=1}^{n-1} s_i$. Ces parts sont toutes aléatoires, et ne donnent aucune information sur le secret à quiconque en possède strictement moins de n. Pour reconstruire le secret, il suffit de calculer $\sum_{i=1}^{n} s_i = \text{secret}$.

Notons que ce partage de secret est homomorphe : en effet, si $s_1, \ldots, s_n$ sont les parts issues d'un secret secret_1 et $r_1, \ldots, r_n$ les parts issues d'un secret secret_2, alors les parts $(s_1 + r_1), \ldots, (s_n + r_n)$ sont issues du secret $(\text{secret}_1 + \text{secret}_2)$.

2.3.2 Partage d'un élément de groupe

Shamir propose un schéma de partage de secret avec un seuil quelconque t en utilisant des polynômes [Sha79]. Le principe de ce schéma est le suivant : un polynôme de degré $t-1$ est complètement déterminé par t de ses points. Ainsi, si le secret est un point particulier de ce polynôme – par exemple la valeur du polynôme en 0 – il est possible de le retrouver en connaissant t points. Cependant, connaître strictement moins de t points ne donne aucune information sur les autres valeurs du polynôme, et donc sur le secret.

Plus formellement, considérons un groupe $\mathbb{G}$ d'ordre p engendré par G, et un secret $S \in \mathbb{G}$. Alors, les parts sont générées de la manière suivante. Tout d'abord, les coefficients du polynôme sont choisis aléatoirement : $A_1, \ldots, A_{t-1} \xleftarrow{R} \mathbb{G}$. Le polynôme utilisé est alors :

$$\begin{array}{rcl} Q : \mathbb{Z}_p & \to & \mathbb{G} \\ z & \mapsto & S \cdot \prod_{j=1}^{t-1} {A_j}^{(z^j)}. \end{array}$$

Cette fonction est bien un polynôme : en notant les logarithmes discrets de tous les éléments dans une base donnée en minuscules, le logarithme discret de ce polynôme est :

$$\begin{array}{rcl} q : \mathbb{Z}_p & \to & \mathbb{Z}_p \\ z & \mapsto & s + \sum_{j=1}^{t-1} a_j \cdot z^j, \end{array}$$

qui est effectivement un polynôme de degré $t-1$, déterminé par la donnée de t de ses points. Les parts sont donc les points du polynômes, c'est-à-dire les

$$(i, S_i) = \Big(i, Q(i)\Big), \quad 1 \leqslant i \leqslant n.$$

Il est ensuite possible de reconstruire le secret, c'est-à-dire $Q(0)$, en utilisant l'interpolation de Lagrange sur t points distincts $\Big\{(i_1, S_{i_1}), \ldots, (i_t, S_{i_t})\Big\}$, c'est-à-dire tels que $i_k \neq i_\ell$ pour $k \neq \ell$:

$$\begin{aligned} S = Q(0) &= \text{Interp}\left(\{(i_j, S_{i_j})\}_{1 \leqslant j \leqslant t}\right) \\ &= \prod_{k=1}^{t} S_{i_k}^{\left(\prod_{\substack{1 \leqslant \ell \leqslant t \\ \ell \neq k}} i_\ell / (i_\ell - i_k)\right)}. \end{aligned}$$

Notons que, à l'instar du partage de secret précédent, ce schéma est homomorphe. De plus il est possible de partager un n-uplet $G = (G_{\mathbf{1}}, \ldots, G_{\mathbf{n}})$ de $\mathbb{G}^n$ en partageant chacun des $G_{\mathbf{k}}$, pour $1 \leqslant k \leqslant n$; il y a alors n polynômes, tous choisis aléatoirement.

Il peut être intéressant pour le prouveur – l'utilisateur qui effectue le partage du secret – de garantir que le secret est bien partagé, c'est-à-dire que le vérifieur – celui qui effectue sa reconstruction – reconstruira le secret attendu. Un schéma de partage de secret ayant cette propriété est dit *vérifiable* [Fel87]. Nous en présentons une instanciation en section 5.1.2 en utilisant le système de preuves de Groth et Sahai [GS08], détaillé en section 2.5.1.

2.4 Schéma d'engagement

2.4.1 Définitions de sécurité

Un schéma d'engagement – parfois appelé mise en gage – est un protocole cryptographique en deux phases qui implique un prouveur $\mathcal{P}$ et un vérifieur $\mathcal{V}$: dans un premier temps, $\mathcal{P}$ s'engage sur un élément en envoyant un engagement à $\mathcal{V}$; $\mathcal{P}$ révèle ensuite l'élément engagé à $\mathcal{V}$ et le convainc que cet élément était masqué dans l'engagement. Le schéma d'engagement considéré nous permet d'utiliser le système de preuves de Groth et Sahai présenté en section 2.5.

Plus précisément, la préparation ComSetup renvoie les paramètres publics params qui permettent aux utilisateurs de s'engager et de vérifier un engagement. Un schéma d'engagement comporte deux autres algorithmes : Com pour s'engager, et ComCheck pour vérifier un engagement. Pour s'engager sur une valeur secrète s, $\mathcal{P}$ calcule un engagement $C_s \leftarrow \mathsf{Com}(\mathsf{params}, s)$, qu'il envoie au vérifieur. Dans un second temps, $\mathcal{P}$ révèle s à $\mathcal{V}$, et le convainc que s est la valeur engagée dans C_s, c'est-à-dire que $\mathsf{ComCheck}(\mathsf{params}, C_s, s) = \mathsf{True}$. Un schéma d'engagement doit garantir deux propriétés ; il doit être *masquant* – connaissant un engagement, il est difficile de retrouver la valeur engagée – et *liant* – une fois engagé, $\mathcal{P}$ ne peut pas changer la valeur s. Ces deux propriétés sont définies à partir des expériences suivantes, où $\mathcal{A}$ est une machine de Turing PPT :

Schéma d'engagement masquant

$\mathsf{params} \leftarrow \mathsf{ComSetup}(\kappa)$
$m_0, m_1 \leftarrow \mathcal{A}(\mathsf{params})$
$b \xleftarrow{R} \{0; 1\}$
$C_{m_b} \leftarrow \mathsf{Com}(\mathsf{params}, m_b)$
$b' \leftarrow \mathcal{A}(C_{m_b})$
Renvoyer 1 si $b' = b$, et 0 sinon.

Schéma d'engagement liant

$\mathsf{params} \leftarrow \mathsf{ComSetup}(\kappa)$
$m_0, m_1, C_m \leftarrow \mathcal{A}(\mathsf{params})$
Renvoyer 1 si :

- $\mathsf{ComCheck}(\mathsf{params}, C_m, m_0) = \mathsf{ComCheck}(\mathsf{params}, C_m, m_1) = \mathsf{True}$,
- $m_0 \neq m_1$,

et 0 sinon.

Dans ces deux cas, l'avantage de $\mathcal{A}$ est noté $\mathrm{Av}_{\mathrm{masquant}}(\kappa)$ et $\mathrm{Av}_{\mathrm{liant}}(\kappa)$; $\mathrm{Av}_{\mathrm{masquant}}(\kappa)$ vaut $|P[b = b'] - 1/2|$ dans le premier jeu, tandis que $\mathrm{Av}_{\mathrm{liant}}(\kappa)$ correspond à la probabilité que le second jeu renvoie 1. Un schéma d'engagement est (a) parfaitement (respectivement calculatoirement) masquant si son avantage associé est nul (resp. négligeable), et (b) parfaitement (resp. calculatoirement) liant si son avantage associé est nul (resp. négligeable).

Dans la suite, nous utilisons implicitement les paramètres publics, et écrivons simplement $\mathsf{Com}(s)$ et $\mathsf{ComCheck}(C_s, s)$.

2.4.2 Engagements SXDH

Nous nous intéressons plus particulièrement aux engagements SXDH proposés par Groth et Sahai [GS08], dont la sécurité repose sur l'hypothèse SXDH. Ce schéma est défini par les algorithmes suivants dans un groupe bilinéaire $\Lambda = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, G_1, G_2)$:

$$\begin{aligned}
\text{ComSetup}_1(\kappa) &= (Y_1 = {G_1}^s, Y_2 = {G_1}^t, Y_3 = {G_1}^{st}), \text{ où } s,t \xleftarrow{R} \mathbb{Z}_p, \\
\text{Com}_1\left(X,(r,r')\right) &= ({G_1}^r \cdot {Y_2}^{r'}, X \cdot {Y_1}^r \cdot {Y_3}^{r'}), \\
\text{ComCheck}_1\left(C_X, X, (r,r')\right) &= \left(C_X \stackrel{?}{=} \text{Com}_1(X,(r,r'))\right)
\end{aligned}$$

dans $\mathbb{G}_1$, et similairement dans $\mathbb{G}_2$ avec G_2, Z_1, Z_2 et Z_3 à la place de G_1, Y_1, Y_2 et Y_3. Il est également possible de s'engager sur un scalaire $x \in \mathbb{Z}_p$ en calculant :

$$\text{Com}_{\mathbb{Z}_p,1}(x,r) = \text{Com}_1\left({G_1}^x,(r,x)\right) \quad \text{ou} \quad \text{Com}_{\mathbb{Z}_p,2}(x,r) = \text{Com}_2\left({G_2}^x,(r,x)\right).$$

En connaissant le paramètre s, il est également possible d'« ouvrir » un engagement pour en extraire soit, dans le cas d'un engagement sur un élément de groupe X, la valeur engagée elle-même, soit, dans le cas d'un engagement sur un scalaire x, l'élément qui était réellement engagé, c'est-à-dire ${G_1}^x$ ou ${G_2}^x$:

$$X = \text{Open}(C_X, s) = C_{X_2} \cdot {C_{X_1}}^{-s}.$$

Dans les deux cas, la valeur extraite détermine uniquement l'élément engagé. Dans la suite, nous appelons s la *clé de trappe* du schéma de l'engagement, et la notons tk.

Ce schéma d'engagement est homomorphe ; en effet, il est possible d'effectuer des opérations sur les engagements. Soient X_1 et X_2 dans $\mathbb{G}_1$, λ, r, r', s, s' des scalaires, et $C_{X_1} = \text{Com}_1\left(X_1,(r,r')\right)$, $C_{X_2} = \text{Com}_1\left(X_2,(s,s')\right)$. Alors, en considérant la multiplication et l'exponentiation composante par composante, on a :

$$\begin{aligned}
C_{X_1} \cdot {C_{X_2}}^{\lambda} &= ({G_1}^r \cdot {Y_2}^{r'}, X_1 \cdot {Y_1}^r \cdot {Y_3}^{r'}) \cdot ({G_1}^s \cdot {Y_2}^{s'}, X_2 \cdot {Y_1}^s \cdot {Y_3}^{s'})^{\lambda} \\
&= ({G_1}^{r+\lambda\cdot s} \cdot {Y_2}^{r'+\lambda\cdot s'}, X_1 \cdot X_2 \cdot {Y_1}^{r+\lambda\cdot s} + \cdot {Y_3}^{r'+\lambda\cdot s'}) \\
&= \text{Com}_1\left(X_1 \cdot X_2^{\lambda}, (r+\lambda\cdot s, r'+\lambda\cdot s')\right).
\end{aligned}$$

Ce schéma est parfaitement liant et calculatoirement masquant sous l'hypothèse SXDH. En utilisant une autre préparation, il est possible d'obtenir un schéma d'engagement parfaitement masquant et calculatoirement liant ; cette préparation est :

$$\text{ComSetup}_1(\kappa) = (Y_1 = {G_1}^s, Y_2 = {G_1}^t, Y_3 = {G_1}^u), \text{ où } s,t,u \xleftarrow{R} \mathbb{Z}_p,$$

et similairement dans $\mathbb{G}_2$; la seule différence est Y_3, où ${G_1}^{st}$ est remplacé par ${G_1}^u$. Dans ce cas, en connaissant en connaissant les paramètres r,s,u ainsi que la valeur engagée et les aléas utilisés, il est possible de révéler des valeurs différentes de la valeur réellement engagée ; dans ce cas, la clé de trappe devient $\text{tk} = (r,s,u)$.

Les préparations parfaitement liantes et parfaitement masquantes sont de plus indistinguables sous l'hypothèse DDH. En pratique, nous utilisons la préparation parfaitement liante, mais la préparation parfaitement masquante est utile pour les preuves.

Pour simplifier les notations, nous écrivons $\mathrm{Com}\left(X,(r,r')\right)$ pour désigner un engagement dans $\mathbb{G}_1$, $\mathbb{G}_2$ ou $\mathbb{Z}_p$ en omettant les indices quand le contexte suffit à les déduire. De plus, quand les aléas utilisés dans un engagement ne sont pas nécessaires pour la compréhension, nous les omettons en écrivant $\mathrm{Com}(X,\_)$. De manière générale, nous notons C_X un engagement sur X.

2.5 Preuves de connaissance à divulgation nulle de connaissance

2.5.1 Définitions de sécurité

Les systèmes de preuves de connaissance à divulgation nulle de connaissance permettent à un prouveur $\mathcal{P}$ de convaincre un vérifieur $\mathcal{V}$ qu'il possède des valeurs – secrètes – qui satisfont des relations données, sans révéler ces valeurs. Dans notre cas, un tel système permet de préserver la vie privée des utilisateurs en masquant les informations identifiantes, tout en garantissant que les calculs sont effectués conformément. Plus précisément, un tel système comporte trois algorithmes [Har11] :

- $\mathrm{Crs}(\kappa)$, un algorithme probabiliste générant une chaîne de référence commune[2] crs, c'est-à-dire les paramètres publics utilisés, qui sont générés par une tierce partie ;
- $\mathrm{Prove}(\mathrm{crs},E_j,x_i)$, un algorithme probabiliste construisant des engagements C_{x_i} sur les secrets x_i ainsi qu'une preuve π que les x_i vérifient les équations E_j ;
- $\mathrm{Verif}(\mathrm{crs},E_j,\pi,C_{x_i})$, un algorithme qui vérifie que la preuve π est valide pour les engagements C_{x_i} et les équations E_j, renvoie True si elle l'est, et False sinon.

Un système de preuves de connaissance à divulgation nulle de connaissance doit garantir trois propriétés : il doit être conforme – une preuve légitime est valide –, sûr – un adversaire ne peut pas construire une preuve illégitime valide –, et à divulgation nulle de connaissance – $\mathcal{V}$ n'apprend rien de la preuve, sinon sa validité. Les deux premières sont définies à partir des expériences suivantes, où $\mathcal{A}$ est une machine de Turing PPT :

Conformité d'un système de preuves

$\mathrm{crs} \leftarrow \mathrm{Crs}(\kappa)$
$(E_j,x_i) \leftarrow \mathcal{A}(\mathrm{crs})$
$\pi, C_{x_i} \leftarrow \mathrm{Prove}(\mathrm{crs},E_j,x_i)$
Renvoyer 1 si :
- les x_i vérifient les équations E_j,
- $\mathrm{Verif}(\mathrm{crs},E_j,\pi,C_{x_i}) = \mathrm{False}$,

et 0 sinon.

2. *Common reference string*

Sûreté d'un système de preuves

$\mathsf{crs} \leftarrow \mathrm{Crs}(\kappa)$
$(E_j, \pi, C_{x_i}) \leftarrow \mathcal{A}(\mathsf{crs})$
Renvoyer 1 si :
- $\mathrm{Verif}(\mathsf{crs}, E_j, \pi) = \mathsf{True}$,
- les x_i ne vérifient pas les E_j, où $x_i = \mathrm{Open}(C_{x_i})$,

et 0 sinon.

La conformité et la sûreté d'un système de preuves sont garanties si la probabilité que ces expériences renvoient 1 est négligeable en fonction du paramètre de sécurité κ. Nous notons $\mathrm{Av}_{\mathrm{NIZK,conformité}}(\kappa)$ et $\mathrm{Av}_{\mathrm{NIZK,sûreté}}(\kappa)$ les avantages associées. La notion de divulgation nulle de connaissance est plus complexe [Gro06]. Brièvement, cette notion implique que si quelqu'un était capable de créer des preuves illégitimes valides, sans connaître la solution des équations, alors un adversaire serait incapable de distinguer une preuve légitime d'une illégitime. Il faut introduire deux nouveaux algorithmes pour définir formellement cette propriété :

- $(\mathsf{crs}, \mathsf{tk}) \leftarrow \mathrm{SimCrs}(\kappa)$, qui génère une crs simulée et une clé de trappe tk ;
- $\pi \leftarrow \mathrm{Sim}(\mathsf{crs}, E_j, C_{x_i}, \mathsf{tk})$ qui génère une preuve simulée sur les équations E_j, valide pour les engagements C_{x_i} ; grâce à la clé de trappe tk, cette preuve peut être construite sans nécessairement en connaître de solution.

Un système de preuves est à divulgation nulle de connaissance si une crs simulée est indistinguable d'une crs classique, et si une preuve simulée est indistinguable d'une preuve légitime ; c'est-à-dire, en considérant un adversaire $\mathcal{A}$, si la probabilité que les deux expériences suivantes renvoient 1 est négligeable en fonction de κ :

Indistinguabilité des crs

$b \xleftarrow{R} \{0; 1\}$
Si $b = 0$, $\mathsf{crs} \leftarrow \mathrm{Crs}(\kappa)$
Sinon, $(\mathsf{crs}, \mathsf{tk}) \leftarrow \mathrm{SimCrs}(\kappa)$
$b' \leftarrow \mathcal{A}(\mathsf{crs})$
Renvoyer 1 si $b = b'$, et 0 sinon.

Indistinguabilité des preuves

$(\mathsf{crs}, \mathsf{tk}) \leftarrow \mathrm{SimCrs}(\kappa)$
$(x_i, E_j) \leftarrow \mathcal{A}(\mathsf{crs})$
$\pi_0, C_{x_i} \leftarrow \mathrm{Prove}(\mathsf{crs}, E_j, x_i)$
$\pi_1 \leftarrow \mathrm{Sim}(\mathsf{crs}, E_j, C_{x_i}, \mathsf{tk})$
$b \xleftarrow{R} \{0, 1\}$
$b' \leftarrow \mathcal{A}(\pi_b, C_{x_i}, \mathsf{tk})$
Renvoyer 1 si :
- les x_i vérifient les équations E_j,
- $b = b'$,

et 0 sinon.

Nous notons $\mathrm{Av}_{\mathrm{crs}}(\kappa)$ et $\mathrm{Av}_{\mathrm{NIZK,indist.}}$ les avantages associés. D'autres notions moins fortes que la divulgation nulle de connaissance existent, comme l'indistinguabilité des témoins, qui garantit que si les équations E_j admettent plusieurs solutions x_i et y_i – des *témoins* –, un adversaire est incapable de savoir laquelle a permis de générer la preuve. Cette propriété ne garantit cependant pas que le vérifieur n'obtient aucune information sur la solution ; par exemple, quand cette solution est unique, l'adversaire pourrait apprendre sa valeur.

2.5.2 Système de preuves de Groth et Sahai

Nous présentons maintenant le système de preuves à divulgation nulle de connaissance proposé par Groth et Sahai [GS08], qui permet de prouver des équations dans les groupes bilinéaires – c'est-à-dire des équations relativement générales, qui peuvent s'appliquer à plusieurs cryptosystèmes –, tout en étant efficacement implémentable. Dans cette section, nous ne considérons que l'instanciation reposant sur le problème SXDH ; il existe d'autres instanciations reposant par exemple sur le problème linéaire décisionnel [BBS04].

Description générale

Le système de preuve de Groth et Sahai [GS08] est un exemple de ces systèmes de preuves, qui permet de calculer des preuves de connaissance non-interactives à divulgation nulle de connaissance (NIZK) dans un groupe bilinéaire $\Lambda = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, G_1, G_2)$ sur quatre types de relations : des équations de produit de couplage (2.1), des équations de multiplications multi-scalaires dans $\mathbb{G}_1$ (2.2) ou dans $\mathbb{G}_2$ (2.3) et des équations de multiplications scalaires (2.4), c'est-à-dire :

$$\left(\prod_{i=1}^{n} e(A_i, V_i)\right) \cdot \left(\prod_{j=1}^{m} e(U_j, B_j)\right) \cdot \left(\prod_{i=1}^{n}\prod_{j=1}^{m} e(U_j, V_i)^{\gamma_{ji}}\right) = e(R_1, R_2), \tag{2.1}$$

$$\left(\prod_{i=1}^{n} {A_i}^{v_i}\right) \cdot \left(\prod_{j=1}^{m} {U_j}^{b_j}\right) \cdot \left(\prod_{i=1}^{n}\prod_{j=1}^{m} {U_j}^{\gamma_{ji}\, v_i}\right) = R_1, \tag{2.2}$$

$$\left(\prod_{i=1}^{n} {V_i}^{a_i}\right) \cdot \left(\prod_{j=1}^{m} {B_j}^{u_j}\right) \cdot \left(\prod_{i=1}^{n}\prod_{j=1}^{m} {V_i}^{\gamma_{ji}\, u_j}\right) = R_2, \tag{2.3}$$

$$\left(\sum_{i=1}^{n} a_i\, v_i\right) + \left(\sum_{j=1}^{m} b_j\, u_j\right) + \left(\sum_{i=1}^{n}\sum_{j=1}^{m} \gamma_{ji}\, u_j\, v_i\right) = r. \tag{2.4}$$

Dans ces équations, R_1, R_2, r, ainsi que les A_i, a_i, B_j, b_j et γ_{ji} sont publics tandis que les U_j, u_j, V_i, v_i sont secrets.

Pour une équation donnée, la preuve NIZK associée repose sur des engagements SXDH sur les valeurs secrètes de cette équation. Comme ce schéma d'engagement est homomorphe, les engagements eux-mêmes vérifient des relations qui dépendent de l'équation initiale. Un prouveur peut alors calculer des éléments de groupe à partir de l'équation et des aléas utilisés dans les engagements. À partir de ces éléments et des valeurs publiques de l'équation, un vérifieur peut vérifier l'équation sans avoir accès aux secrets. Ainsi, la crs du système de preuves correspond

aux paramètres des schéma d'engagement dans les groupes $\mathbb{G}_1$ et $\mathbb{G}_2$. Lorsque le schéma d'engagement SXDH est parfaitement liant, un engagement correspond à une unique valeur, qui peut être obtenue avec la clé de trappe. Ainsi, les relations vérifiées par les engagements sont également vérifiées par les valeurs correspondantes ; une preuve est donc valide si et seulement si les valeurs engagées vérifient les équations. D'un autre côté, lorsque le schéma d'engagement SXDH est parfaitement masquant, la clé de trappe permet de faire correspondre n'importe quelle valeur à un engagement. En connaissant cette clé de trappe, il est donc possible de construire des preuves que des secrets vérifient une équation sans que ces secrets la vérifient effectivement. En pratique, comme pour les engagements SXDH, nous utilisons la préparation parfaitement liante.

Le schéma proposé par Groth et Sahai est *non-interactif* puisque le prouveur (et respectivement le vérifieur) calculent la preuve (respectivement la vérifient) sans interagir. Il est de plus possible de combiner plusieurs relations pour prouver qu'un secret vérifie plusieurs équations, en réutilisant l'engagement sur ce secret. Dans la suite, nous notons les preuves de connaissance en utilisant la notation introduite par Camenisch et Stadler [CS97] ; c'est-à-dire qu'une preuve que les secrets $(u_1, \ldots, u_k)$ vérifient les équations $(E_1, \ldots, E_\ell)$ est notée

$$\text{NIZK}\{u_1, \ldots, u_k : E_1 \wedge \cdots \wedge E_\ell\}.$$

Présentation détaillée

Nous présentons maintenant la construction d'une preuve NIZK sur une équation de produits de couplage ; les preuves sur les autres types d'équation sont similaires. La préparation de ce schéma consiste à mettre en place le schéma d'engagement SXDH dans $\mathbb{G}_1$ ainsi que dans $\mathbb{G}_2$; ainsi, la chaîne de référence commune crs est le groupe bilinéaire Λ, $Y_1, Y_2, Y_3 \in \mathbb{G}_1$ et $Z_1, Z_2, Z_3 \in \mathbb{G}_2$.

Nous considérons le cas où un prouveur $\mathcal{P}$ connaît $U_1, \ldots, U_m \in \mathbb{G}_1$ et $V_1, \ldots, V_n \in \mathbb{G}_2$ tels que :

$$\left(\prod_{i=1}^{n} e(A_i, V_i)\right) \cdot \left(\prod_{j=1}^{m} e(U_j, B_j)\right) \cdot \left(\prod_{i=1}^{n}\prod_{j=1}^{m} e(U_j, V_i)^{\gamma_{ji}}\right) = e(R_1, R_2), \tag{2.5}$$

où les A_i sont des éléments de $\mathbb{G}_1$, les B_j des éléments de $\mathbb{G}_2$, les γ_{ji} sont des scalaires et $(R_1, R_2) \in \mathbb{G}_1 \times \mathbb{G}_2$. Pour prouver qu'il connaît les U_j et les V_i satisfaisant l'équation (2.5), $\mathcal{P}$ calcule des engagements SXDH (voir section 2.4.2) sur les U_j et les V_i. Soient r_{j1}, r_{j2}, s_{i1}, s_{i2} les aléas de $\mathbb{Z}_p$ utilisés dans ces engagements :

$$(\mathcal{U}_{j1}, \mathcal{U}_{j2}) = \text{Com}\left(U_j, (r_{j1}, r_{j2})\right)$$
$$(\mathcal{V}_{i1}, \mathcal{V}_{i2}) = \text{Com}\left(V_i, (s_{i1}, s_{i2})\right).$$

Le prouveur choisit ensuite quatre aléas t_{11}, t_{12}, t_{21}, t_{22} dans $\mathbb{Z}_p$, et calcule les engagements

suivants : [3]

$$(\pi_{11},\pi_{12}) = \mathrm{Com}\left(\prod_j \left(B_j \cdot \prod_i V_i^{\gamma_{ji}}\right)^{r_{j1}}, \left(\vec{r_1}\Gamma\vec{s_1}^{\,\intercal} - t_{11}, \vec{r_1}\Gamma\vec{s_2}^{\,\intercal} - t_{21}\right)\right)$$

$$(\pi_{21},\pi_{22}) = \mathrm{Com}\left(\prod_j \left(B_j \cdot \prod_i V_i^{\gamma_{ji}}\right)^{r_{j2}}, \left(\vec{r_2}\Gamma\vec{s_1}^{\,\intercal} - t_{12}, \vec{r_2}\Gamma\vec{s_2}^{\,\intercal} - t_{22}\right)\right)$$

$$(\theta_{11},\theta_{12}) = \mathrm{Com}\left(\prod_i \left(A_i \cdot \prod_j U_j^{\gamma_{ji}}\right)^{s_{i1}}, \left(t_{11}, t_{12}\right)\right)$$

$$(\theta_{21},\theta_{22}) = \mathrm{Com}\left(\prod_i \left(A_i \cdot \prod_j U_j^{\gamma_{ji}}\right)^{s_{i2}}, \left(t_{21}, t_{22}\right)\right)$$

où

- $\vec{r_k} = (r_{1k}, \ldots, r_{mk})$ pour $k \in \{1;2\}$,
- Γ est la matrice $m \times n$ composée des γ_{ji},
- et $\vec{s_\ell} = (s_{1\ell}, \ldots, s_{n\ell})$ pour $\ell \in \{1;2\}$,

c'est-à-dire, en notant $\intercal$ la transposée d'une matrice, $\vec{r_k}\Gamma\vec{s_\ell}^{\,\intercal} = \sum_i \sum_j r_{jk}\gamma_{ji}s_{i\ell}$.

Les $\mathcal{U}_{jk}$ et $\mathcal{V}_{i\ell}$ sont les engagements sur les secrets, et les $\pi_{k\ell}$ et $\theta_{k\ell}$ constituent la preuve que les valeurs masquées dans ces engagements satisfont l'équation (2.5). Pour vérifier cette preuve, $\mathcal{V}$ vérifie que les quatre équations suivantes sont valides :

$$e(G_1,\pi_{11})\, e(Y_2,\pi_{21})\, e(\theta_{11},G_2)\, e(\theta_{21},Z_2) = \left(\prod_i e\Big(\prod_j \mathcal{U}_{j1}^{\gamma_{ji}}, \mathcal{V}_{i1}\Big)\right) \tag{2.6}$$

$$e(G_1,\pi_{12})\, e(Y_2,\pi_{22})\, e(\theta_{11},Z_1)\, e(\theta_{21},Z_3) = \left(\prod_j e\Big(\mathcal{U}_{j1}, B_j \cdot \prod_i \mathcal{V}_{i2}^{\gamma_{ji}}\Big)\right) \tag{2.7}$$

$$e(Y_1,\pi_{11})\, e(Y_3,\pi_{21})\, e(\theta_{12},G_2)\, e(\theta_{22},Z_2) = \left(\prod_i e\Big(A_i \cdot \prod_j \mathcal{U}_{j2}^{\gamma_{ji}}, \mathcal{V}_{i1}\Big)\right) \tag{2.8}$$

$$\begin{aligned} e(R_1,R_2)\, e(Y_1,\pi_{12})\, e(Y_3,\pi_{22})\, e(\theta_{12},Z_1)\, e(\theta_{22},Z_3) = {} & \left(\prod_i e\Big(A_i \cdot \prod_j \mathcal{U}_{j2}^{\gamma_{ji}}, \mathcal{V}_{i1}\Big)\right) \\ & \cdot \left(\prod_j e\Big(\mathcal{U}_{j2}, B_j\Big)\right) \end{aligned} \tag{2.9}$$

où Y_1, Y_2, Y_3 correspondent à la préparation du schéma d'engagement dans $\mathbb{G}_1$, et Z_1, Z_2, Z_3 à sa préparation dans $\mathbb{G}_2$ (voir section 2.4.2).

Si ces équations sont effectivement valides, alors $\mathcal{V}$ est convaincu que $\mathcal{P}$ connaît les U_j et les V_i satisfaisant l'équation (2.5). Groth et Sahai prouvent que ce schéma est un système de preuves à divulgation nulle de connaissance [GS08].

3. Pour simplifier les notations, les produits et sommes se font pour $1 \leqslant j \leqslant m$ et $1 \leqslant i \leqslant n$.

Optimisation de la vérification

Le calcul des couplages dans un groupe bilinéaire est très coûteux : par exemple, dans le groupe bilinéaire proposé par Aranha et coll. [Ara+11], le calcul d'un couplage est six fois plus coûteux qu'une exponentiation dans $\mathbb{G}_1$, et deux fois plus coûteux qu'une exponentiation dans $\mathbb{G}_2$. Comme la vérification d'une preuve NIZK nécessite de calculer de nombreux couplages – $3n+2m+17$ pour la vérification présentée précédemment –, une telle vérification est également très coûteuse. Blazy et coll. [Bla+10] proposent une méthode pour grouper les vérifications des preuves et réduire le nombre de couplages nécessaires à $2n + m + 9$ en exploitant la bilinéarité du couplage.

Leur idée est de mettre les parties gauches et droites des équations (2.6) à (2.9) à une puissance $r_i \xleftarrow{\mathbb{R}} \mathbb{Z}_p$, et de vérifier que l'équation $(2.6)^{r_1} \cdot (2.7)^{r_2} \cdot (2.8)^{r_3} \cdot (2.9)^{r_4}$ est toujours valide. Il est alors possible d'exploiter la bilinéarité du couplage pour déplacer les produits à l'intérieur des couplages, ce qui transforme les produits de couplage en produits dans les groupes $\mathbb{G}_1$ ou $\mathbb{G}_2$. Le côté gauche de cette équation est :

$$\begin{aligned} & e({R_1}^{r_4}, R_2) \\ \cdot\, & e({G_1}^{r_1} \cdot {Y_1}^{r_3}, \pi_{11}) \cdot e({Y_2}^{r_1} \cdot {Y_3}^{r_3}, \pi_{21}) \cdot e({\theta_{11}}^{r_1} \cdot {\theta_{12}}^{r_3}, G_2) \cdot e({\theta_{21}}^{r_1} \cdot {\theta_{22}}^{r_3}, Z_2) \\ \cdot\, & e({G_1}^{r_2} \cdot {Y_1}^{r_4}, \pi_{12}) \cdot e({Y_2}^{r_2} \cdot {Y_3}^{r_4}, \pi_{22}) \cdot e({\theta_{11}}^{r_2} \cdot {\theta_{12}}^{r_4}, Z_1) \cdot e({\theta_{21}}^{r_2} \cdot {\theta_{22}}^{r_4}, Z_3), \end{aligned} \tag{2.10}$$

tandis que le côté droit est :

$$\begin{aligned} & \prod_{i=1}^{n} e\Big(\big(\textstyle\prod_{j=1}^{m} {\mathcal{U}_{j1}}^{\gamma_{ji}}\big)^{r_1} \cdot \big(A_i \cdot \textstyle\prod_{j=1}^{m} {\mathcal{U}_{j2}}^{\gamma_{ji}}\big)^{r_3}, \mathcal{V}_{i1}\Big) \\ \cdot\, & \prod_{i=1}^{n} e\Big(\big(\textstyle\prod_{j=1}^{m} {\mathcal{U}_{j1}}^{\gamma_{ji}}\big)^{r_2} \cdot \big(A_i \cdot \textstyle\prod_{j=1}^{m} {\mathcal{U}_{j2}}^{\gamma_{ji}}\big)^{r_4}, \mathcal{V}_{i2}\Big) \\ \cdot\, & \prod_{j=1}^{m} e\big({\mathcal{U}_{j1}}^{r_1} \cdot {\mathcal{U}_{j2}}^{r_2}, B_j\big). \end{aligned} \tag{2.11}$$

Une preuve NIZK est alors valide si et seulement si (2.10) = (2.11).

Si plusieurs preuves NIZK impliquent les mêmes secrets, il est possible de les vérifier simultanément en exploitant la même technique pour réduire encore plus le nombre de couplages nécessaires.

2.6 Schémas de signature

Un schéma de signature permet à un utilisateur de prouver qu'il est à l'origine d'un message. Dans cette section, nous présentons trois schémas de signature différents. Nous commençons par un schéma de signature classique [KW03], qui est efficace et possède une réduction de sécurité fine au problème CDH. Le second est un schéma de signature *automorphe* [Abe+10] ; un tel schéma utilise uniquement les opérations arithmétiques d'un groupe bilinéaire, et est donc complètement compatible avec le système de preuves de Groth et Sahai présenté précédemment. Nous combinons ces deux outils pour construire un schéma de signature *proxy*

anonyme [FP08], qui permet aux signataires de déléguer la capacité de signer des messages en leur nom à d'autres utilisateurs – appelés mandataires – sans qu'un mandataire signant un message ne révèle son identité ; un mandataire peut déléguer cette capacité à d'autres utilisateurs et créer un nombre arbitraire de niveaux de délégation. Les signatures proxy anonymes sont une combinaison des signatures de groupe [CH91] et des signatures proxy [MUO96] ; les premières permettent aux membres d'un groupe de signer en son nom sans révéler leur identité – sans permettre de délégation à plusieurs niveaux –, tandis que les secondes permettent aux utilisateurs de déléguer la capacité de signer, sans préserver la vie privée des mandataires. Nous utilisons ce schéma au chapitre 5 pour permettre aux clients de générer des pseudonymes qui signeront en leur nom, tout en garantissant l'inassociabilité de ces pseudonymes et en masquant leur identité.

Dans cette section, nous commençons par présenter la spécification et les propriétés de sécurité des signatures classiques, puis détaillons le schéma de signature automorphe proposé par Abe et coll. [Abe+10], avant de présenter les propriétés de sécurité des signatures proxy anonymes définies par Fuchsbauer et Pointcheval [FP08], ainsi que leur instanciation proposée par Abe et coll. [Abe+10].

2.6.1 Schéma de signature classique

Un schéma de signature classique comporte trois algorithmes :

- $\mathrm{Gen}(\kappa)$, qui génère aléatoirement une paire de clés de signature $(\mathsf{sk}, \mathsf{vk})$ en fonction du paramètre de sécurité : une clé de signature, privée, ainsi qu'une clé de vérification, publique.
- $\mathrm{Sign}(\mathsf{sk}, M)$, qui construit une signature σ portant sur le message M à partir de la clé privée sk ; cet algorithme peut être randomisé.
- $\mathrm{Verif}(\sigma, M, \mathsf{vk})$, qui vérifie si une signature σ sur un message M est valide pour la clé de vérification vk ; dans ce cas, cet algorithme renvoie True, et False dans le cas contraire.

Une propriété de sécurité classique des schémas de signature est l'inforgeabilité existentielle face à des attaques adaptatives par messages choisis (EU-CMA). Pour définir cette propriété, l'adversaire a accès à un *oracle de signature*, que nous notons $O_{\mathsf{sk}}(\cdot)$. L'adversaire peut faire appel à cet oracle sur n'importe quel message pour obtenir une signature valide pour la paire de clé considérée. Tous les messages pour lesquels $\mathcal{A}$ a demandé une signature sont enregistrés dans l'ensemble Q. La propriété EU-CMA est formellement définie à partir de l'expérience suivante, où $\mathcal{A}$ est une machine de Turing PPT :

Schéma de signature EU-CMA

$(\mathsf{vk}, \mathsf{sk}) \leftarrow \mathrm{Gen}(\kappa)$
$(M, \sigma) \leftarrow \mathcal{A}^{O_{\mathsf{sk}}(\cdot)}(\mathsf{vk})$
Renvoyer 1 si $\mathrm{Verif}(\sigma, M, \mathsf{vk})$ et si $M \notin Q$

L'avantage de $\mathcal{A}$, noté $\mathrm{Av}_{\text{EU-CMA}}(\kappa)$, correspond à la probabilité que cette expérience renvoie 1. Un schéma de signature est dit EU-CMA si cet avantage est négligeable.

Il existe de nombreux schéma de signature, comme par exemple DSA et sa variante sur courbes elliptiques ECDSA [Nat13] ou le schéma de signature de Schnorr [Sch89 ; Sch91]. Dans

la suite, nous considérons le schéma de signature proposé par Katz et Wang [KW03], qui repose sur l'hypothèse CDH, c'est-à-dire une hypothèse similaire à celle du schéma d'engagement SXDH et du système de preuves de Groth et Sahai [GS08] présentés précédemment ; de plus, la réduction de sécurité de ce schéma est *fine*, c'est-à-dire que forger une signature avec ce schéma est au moins aussi difficile que de résoudre le problème CDH. Finalement, ce schéma est efficace puisque la construction d'une signature requiert deux exponentiations dans le groupe considéré, qui peuvent être précalculées, tandis que la vérification d'une signature en requiert quatre.

Schéma de signature de Katz-Wang

Le schéma de signature de Katz et Wang [KW03] repose sur un groupe $\mathbb{G}$ d'ordre p premier. Pour générer une paire de clés, un utilisateur choisit deux générateurs G_1 et G_2 de ce groupe, ainsi qu'un nombre aléatoire $x \xleftarrow{R} \mathbb{Z}_p$, qui donne la clé secrète ; la clé publique est le quadruplet $(G_1, G_2, Y_1 = {G_1}^x, Y_2 = {G_2}^x)$.

Ce schéma permet de signer des messages de taille quelconque grâce à une fonction de hachage $H : \{0,1\}^* \to \mathbb{Z}_p$. Pour signer un message M, le signataire choisit $r \xleftarrow{R} \mathbb{Z}_p$ et calcule $A = {G_1}^r$, $B = {G_2}^r$, $c = H(\mathsf{vk}, A, B, M)$ et $s = c \cdot \mathsf{sk} + r$. La signature finale est

$$\mathrm{Sign}(\mathsf{sk}, M, r) = (c, s) = \Big(H(\mathsf{vk}, {G_1}^r, {G_2}^r, M), \quad H(\mathsf{vk}, {G_1}^r, {G_2}^r, M) \cdot \mathsf{sk} + r\Big).$$

Pour vérifier la validité d'une signature $\sigma = (s, c)$ pour un message M et une clé de vérification $\mathsf{vk} = (G_1, G_2, Y_1, Y_2)$, il suffit de calculer $A' = {G_1}^s \cdot {Y_1}^{-c}$ et $B' = {G_2}^s \cdot {Y_2}^{-c}$, et de vérifier que

$$c = H(\mathsf{vk}, A', B', M).$$

2.6.2 Signatures automorphes

Les schémas de signature classiques reposent généralement sur le paradigme *hash-and-sign* : pour être signés, les messages sont préalablement hachés, ce qui permet de signer des messages de taille quelconque. Cependant, comme présenté précédemment, le système de preuves de Groth et Sahai ne permet de prouver que des équations dans un groupe bilinéaire, c'est-à-dire que ce système n'est pas compatible avec les fonctions de hachage. Il n'est donc pas possible de combiner signatures classiques et preuves de Groth-Sahai. Pour cette raison, Abe et coll. [Abe+10] proposent un schéma de signature où les clés de signature, les messages et les signatures sont des éléments de groupe bilinéaire, et où la vérification d'une signature consiste en des équations de produits de couplage (voir équation 2.1) ; un tel schéma de signature est dit *automorphe*[4].

Pour distinguer les schémas de signature automorphes des schémas de signature classiques, nous notons SPSign et SPVerif ses opérations de signature et de vérification ; pour le schéma de signature classiques, elles sont notées Sign et Verif.

Le schéma de signature automorphe proposé par Abe et coll. repose sur un groupe bilinéaire $\Lambda = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, G_1, G_2)$. Ce schéma requiert également trois éléments de groupe choisis aléatoirement : $F, K, T \xleftarrow{R} \mathbb{G}_1$.

4. ou parfois *Structure-Preserving*.

La paire de clés d'un utilisateur $\mathcal{U}$ comporte une clé de signature $\mathsf{sk}_\mathcal{U} = u \in \mathbb{Z}_p$ et une clé de vérification $\mathsf{vk}_\mathcal{U} = ({G_1}^{\mathsf{sk}_\mathcal{U}}, {G_2}^{\mathsf{sk}_\mathcal{U}})$. Pour signer un message $M \in \mathbb{G}_1$, $\mathcal{U}$ choisit deux aléas $\rho, \gamma \xleftarrow{R} \mathbb{Z}_p$ et calcule :

$$\mathrm{SPSign}\left(\mathsf{sk}_\mathcal{U}, M, (\rho, \gamma)\right) = (A_M, C_M, D_M, R_M, S_M),$$

où

$$A_M = (K \cdot T^\rho \cdot M)^{\frac{1}{u+\gamma}}$$

$$C_M = F^\gamma \qquad D_M = {G_2}^\gamma$$
$$R_M = {G_1}^\rho \qquad S_M = {G_2}^\rho.$$

Notons que A_M est le seul élément de la signature dépendant de la clé de signature ou du message ; C_M, D_M, R_M et S_M dépendent uniquement des paramètres ainsi que des aléas choisis.

Une signature $\sigma = (A_M, C_M, D_M, R_M, S_M)$ est valide pour un message M et une clé de vérification $\mathsf{vk}_\mathcal{U} = (V_\mathcal{U}, W_\mathcal{U})$ si et seulement si les équations suivantes sont vérifiées :

$$e(A_M, W_\mathcal{U} \cdot D_M) = e(K \cdot M, G_2)\, e(T, S_M) \tag{2.12}$$
$$e(C_M, G_2) = e(F, D_M) \tag{2.13}$$
$$e(R_M, G_2) = e(G_1, S_M); \tag{2.14}$$

nous écrivons $\mathrm{SPVerif}(\sigma, M, \mathsf{vk}_\mathcal{U}) = \mathrm{True}$ si tel est est le cas.

Il est également possible de signer un scalaire $m \in \mathbb{Z}_p$ au lieu de $M \in \mathbb{G}_1$ en signant ${G_1}^m$ à la place de M. Cependant, pour signer un élément $M_2 \in G_2$, il est nécessaire de connaître $M_1 \in \mathbb{G}_1$ tel que $M_1 = {G_1}^{\log_{G_2} M_2}$; il faut ensuite signer M_1 et montrer que M_1 et M_2 ont le même logarithme discret, c'est-à-dire que $e(M_1, G_2) = e(G_1, M_2)$.

Comme les clés de vérifications sont des messages acceptables, il est possible de signer une clé de vérification pour créer un certificat. Soit $\mathcal{C}$ l'autorité de certification ayant la paire de clé $(\mathsf{sk}_\mathcal{C}, \mathsf{vk}_\mathcal{C})$, où $\mathsf{vk}_\mathcal{C} = (V_\mathcal{C}, W_\mathcal{C})$. Pour certifier une clé de vérification $\mathsf{vk}_\mathcal{U} = (V_\mathcal{U}, W_\mathcal{U})$, l'autorité signe $V_\mathcal{U}$:

$$\mathsf{cert}_\mathcal{U} = \mathrm{Cert}\left(\mathsf{vk}_\mathcal{U}, \mathsf{sk}_\mathcal{C}, (r, s)\right) = \mathrm{SPSign}\left(\mathsf{sk}_\mathcal{C}, V_\mathcal{U}, (r, s)\right) = (A_\mathcal{U}, C_\mathcal{U}, D_\mathcal{U}, R_\mathcal{U}, S_\mathcal{U})$$

Un tel certificat est valide si et seulement si les équations suivantes sont vérifiées :

$$e(A_\mathcal{U}, W_\mathcal{C} \cdot D_\mathcal{U}) = e(K \cdot V_\mathcal{U}, G_2)\, e(T, S_\mathcal{U}) \tag{2.15}$$
$$e(C_\mathcal{U}, G_2) = e(F, D_\mathcal{U}) \tag{2.16}$$
$$e(R_\mathcal{U}, G_2) = e(G_1, S_\mathcal{U}) \tag{2.17}$$
$$e(V_\mathcal{U}, G_2) = e(G_1, W_\mathcal{U}). \tag{2.18}$$

Notons que les équations (2.15) à (2.17) sont identiques aux équations (2.12) à (2.14) en remplaçant $W_\mathcal{C}$ par $W_\mathcal{U}$. L'équation (2.18) vérifie que la clé $\mathsf{vk}_\mathcal{U}$ est valide.

Les auteurs montrent également comment signer un vecteur de messages à partir de ce schéma de signature. Plus précisément, une signature automorphe sur deux messages consiste

en trois signatures automorphes, tandis qu'une une signature sur n messages, $n \geqslant 3$, consiste en $3 \cdot n$ signatures. Nous renvoyons à l'article de Abe et coll. pour plus de détails sur de telles signatures [Abe+10]. Une signature sur un vecteur de message $M_1, \ldots, M_n$ est notée :

$$\text{SPSign}\left(\mathsf{sk}, \langle M_1, \ldots, M_n \rangle\right).$$

Les auteurs montrent que ce schéma est fortement inforgeable face à des attaques à messages choisis – c'est-à-dire que, même en ayant à sa disposition un oracle lui permettant de générer des signatures sur des messages quelconques, un adversaire ne peut forger une signature sur un nouveau message – en se reposant sur deux hypothèses, dont une est dérivée du DDH. Dans la suite, nous notons $\mathrm{Av}_{\mathrm{SP},\text{EU-CMA}}(\kappa, \ell)$ l'avantage de l'adversaire pour forger une signature automorphe sur un message de longueur ℓ.

2.6.3 Signatures proxy anonymes

Spécification et propriétés de sécurité

Un schéma de signature proxy anonyme est un schéma de signature qui permet à un utilisateur de déléguer sa signature à un autre utilisateur en lui conférant un mandat, tout en préservant la vie privée des mandataires. Fuchsbaueur et Pointcheval [FP08] définissent un tel schéma à partir de cinq algorithmes :

- Setup(κ), qui génère les paramètres publics, la clé de l'émetteur ik ainsi que la clé de certificat de l'autorité d'ouverture ock – cette clé permet d'utiliser Open pour apprendre le signataire d'une signature ; pour s'enregistrer, un utilisateur génère une paire de clés $(\mathsf{vk}_u, \mathsf{sk}_u)$ et reçoit un certificat cert sur vk_u de l'émetteur.
- Delegate, qui prend sk_x, vk_y et éventuellement un mandat conféré à x en argument ; cet algorithme génère un mandat de x sur y, permettant à y de signer des messages pour x, et éventuellement pour l'utilisateur ayant conféré un mandat à x.
- PSig, qui prend sk_y, un message M ainsi qu'un mandat conféré à y provenant de x – ce mandat peut être délégué par de multiples utilisateurs entre x et y –, et renvoie une signature σ.
- PVerif, qui prend vk_x, un message M et une signature σ en argument, et renvoie True ou False.
- Open, qui prend la clé d'ouverture ok_x associée à x – générée à partir de la clé de l'autorité d'ouverture ock –, une signature σ et un message M, et renvoie le signataire final ainsi que la chaîne d'utilisateurs lui ayant délégué le mandat.

Les auteurs définissent également les quatre propriétés de sécurité qu'un schéma de signature proxy anonyme doit garantir ; informellement, ces propriétés sont les suivantes :

Conformité Un signataire honnête ayant reçu une délégation légitime est capable d'émettre des signatures valides.

Anonymat Étant donnés un message, un utilisateur x ayant mandaté deux utilisateurs honnêtes y_0 et y_1, les signatures de y_0 et de y_1 au nom de x sont indistinguables.

Traçabilité Tout signature valide peut être ouverte par l'autorité d'ouverture.

Fausses accusations Un utilisateur honnête ne peut être faussement accusé d'avoir délégué un mandat ou d'avoir signé un message sous le mandat d'un utilisateur pour lequel il n'a pas reçu de mandat.

Notons qu'en empêchant les fausses accusations, un schéma de signature proxy anonyme garantit également l'inforgeabilité des signatures.

Instanciation

Le schéma de signature proxy anonyme instancié par Abe et coll. [Abe+10] repose sur les signatures automorphes ainsi que sur le système de preuve de Groth et Sahai [GS08] (voir section 2.5.2). Sa préparation repose donc sur celle du système de preuve et celle des signatures automorphes. Dans la suite, nous considérons un groupe bilinéaire $\Lambda = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, G_1, G_2)$, les éléments permettant de s'engager dans $\mathbb{G}_1$ – Y_1, Y_2 et Y_3 – et dans $\mathbb{G}_2$ – Z_1, Z_2 et Z_3 –, ainsi que ceux nécessaires aux signatures automorphes – F, K et T.

Pour s'enregistrer un utilisateur $\mathcal{U}$ génère une paire de clés de signature automorphes $\left(\mathsf{sk}_{\mathcal{U}}, \mathsf{vk}_{\mathcal{U}} = (V_{\mathcal{U}}, W_{\mathcal{U}})\right)$ et reçoit un mandat de l'émetteur C, c'est-à-dire un certificat $\mathsf{cert}_{\mathcal{U}}$:

$$\mathsf{cert}_{\mathcal{U}} = \mathsf{Cert}(\mathsf{vk}_{\mathcal{U}}, \mathsf{sk}_C, \_) = (A_{\mathcal{U}}, C_{\mathcal{U}}, D_{\mathcal{U}}, R_{\mathcal{U}}, S_{\mathcal{U}})$$

L'utilisateur $\mathcal{U}$ est alors capable de signer des messages au nom de C sans révéler ni son certificat, ni sa clé de vérification. Pour signer un message M, $\mathcal{U}$ calcule les éléments suivants :

- une signature automorphe sur M : $\sigma = \mathsf{SPSign}(\mathsf{sk}_{\mathcal{U}}, M, \_) = (A_M, C_M, D_M, R_M, S_M)$;
- un engagement sur sa clé de vérification : $C_{\mathsf{vk}_{\mathcal{U}}} = \mathsf{Com}(\mathsf{vk}_{\mathcal{U}}, \_) = (C_{V_{\mathcal{U}}}, C_{W_{\mathcal{U}}})$;
- un engagement sur son certificat : $C_{\mathsf{cert}_{\mathcal{U}}} = \mathsf{Com}(\mathsf{cert}_{\mathcal{U}}, \_) = (C_{A_{\mathcal{U}}}, C_{C_{\mathcal{U}}}, C_{D_{\mathcal{U}}}, C_{R_{\mathcal{U}}}, C_{S_{\mathcal{U}}})$;
- un engagement sur le seul élément de la signature dépendant de sa clé de signature, A_M : $C_{A_M} = \mathsf{Com}(A_M, \_)$;[5]
- une preuve que le certificat engagé dans $C_{\mathsf{cert}_{\mathcal{U}}}$ est valide pour la clé engagée dans $C_{\mathsf{vk}_{\mathcal{U}}}$:

$$\pi_{\mathsf{cert}_{\mathcal{U}}} = \textsc{nizk}\Big\{\mathsf{cert}_{\mathcal{U}}, \mathsf{vk}_{\mathcal{U}} : \mathsf{SPVerif}(\mathsf{cert}_{\mathcal{U}}, \mathsf{vk}_{\mathcal{U}}, \mathsf{vk}_C) \wedge e(V_{\mathcal{U}}, G_2) = e(G_1, W_{\mathcal{U}})\Big\};$$

- une preuve que la signature partiellement engagée dans C_{A_M} est valide pour la clé engagée dans $\mathsf{vk}_{\mathcal{U}}$ et le message M :

$$\pi_\sigma = \textsc{nizk}\Big\{A_M, W_{\mathcal{U}} : e(A_M, W_{\mathcal{U}} \cdot D_M) = e(K \cdot M, G_2) \cdot e(T, S_M)\Big\}.$$

5. Le schéma de signature automorphe ne garantit pas l'indistinguabilité des clés, c'est-à-dire qu'il est peut-être possible de savoir si deux signatures ont été émises par le même signataire ; il faut donc masquer l'élément qui dépend de la clé de signature.

Finalement, une signature proxy anonyme de l'utilisateur $\mathcal{U}$ sur le message M est le multiplet

$$(C_{\mathsf{vk}_\mathcal{U}}, C_{\mathsf{cert}_\mathcal{U}}, C_{A_M}, C_M, D_M, R_M, S_M, \pi_{\mathsf{cert}_\mathcal{U}}, \pi_\sigma).$$

Une telle signature est valide si :

- $\pi_{\mathsf{cert}_\mathcal{U}}$ est valide pour les engagements $C_{\mathsf{cert}_\mathcal{U}}$ et $C_{\mathsf{vk}_\mathcal{U}}$, et pour les éléments publics vk_C, G_1, G_2, F, K, T, M ;
- π_σ est valide pour les engagements C_{A_M} et $C_{W_\mathcal{U}}$, et pour les éléments publics D_M, S_M, K, T, M, G_2 ;
- les équations (2.13) et (2.14) sont valides pour C_M, D_M, R_M, S_M, F, G_1 et G_2.

Un utilisateur peut construire plusieurs signatures proxy anonymes avec les mêmes engagements sur sa clé de vérification et sur son certificat, ce qui permet d'associer ces signatures. Dans ce cas, il suffit de vérifier la preuve $\pi_{\mathsf{cert}_\mathcal{U}}$ pour la première signature, et de vérifier uniquement les π_σ des signatures ultérieures. Dans la suite, nous notons $\mathrm{APSign}(\mathsf{sk}_\mathcal{U}, M)$ la signature et la preuve π_σ, c'est-à-dire :

$$\varsigma = \mathrm{APSign}(\mathsf{sk}_\mathcal{U}, M) = (C_{A_M}, C_M, D_M, R_M, S_M, \pi_\sigma),$$

et nous noterons $\mathrm{APVerif}(\varsigma, M, C_{\mathsf{vk}_\mathcal{U}}) = \mathsf{True}$ si les éléments contenus dans la signature ς sont compatibles avec l'engagement $C_{\mathsf{vk}_\mathcal{U}}$ et le message M.

Il est possible de construire une signature proxy anonyme à plusieurs niveaux de délégations en construisant une chaîne de certificats, et de construire des preuves NIZK prouvant qu'une telle chaîne est valide sans la révéler.

Comme ce schéma repose sur le schéma de signature automorphe de Abe et coll. [Abe+10] et sur le système de preuves de Groth et Sahai [GS08], il est également inforgeable face à des attaques par messages choisis, avec un avantage $\mathrm{Av}_{\mathrm{AP},\text{EU-CMA}}(\kappa, \ell)$ négligeable, et à divulgation nulle de connaissance avec un avantage $\mathrm{Av}_{\mathrm{AP},\mathrm{indist.}}(\kappa)$ qui est nul ou négligeable suivant la crs employée. [6]

2.6.4 Signatures automorphes et proxy anonymes

Le schéma de signature proxy anonyme permet de masquer l'utilisateur ayant effectivement construit une signature, pour ne révéler que le délégataire. Cependant, le système de preuves de Groth et Sahai permet également de masquer tout ou partie des messages signés, ainsi que les signatures elles-mêmes. Dans la suite, nous parlerons de signatures proxy anonymes uniquement dans le premier cas, c'est-à-dire quand les signatures et messages *ne* sont *pas* masqués. Dans les autres cas, nous parlerons de signatures automorphes ainsi que de preuves NIZK.

Pour différencier les signatures classiques et automorphes des signatures proxy anonymes, nous utilisons deux graphies distinctes de la lettre grecque sigma : les première sont notées avec la graphie classique, σ, tandis que les signatures proxy anonymes sont notées avec la graphie finale, ς.

6. Dans ces avantages, « AP » signifie *Anonymous Proxy*.

2.7 Résumé des outils cryptographiques

Dans ce chapitre, nous avons présenté divers outils cryptographiques existants. Nous avons commencé par rappeler des notions aussi bien d'algèbre que de cryptographie permettant de comprendre le fonctionnement de ces outils. Pour chaque schéma cryptographique présenté, nous avons décrit les propriétés de sécurité garanties ainsi que les hypothèses sur lesquelles ce schéma s'appuie.

Le premier schéma décrit est le partage de secret. Ce schéma permet de partager une valeur parmi un ensemble d'utilisateurs, de manière à ce qu'une collusion de petite taille ne puisse rien savoir du secret, tout en garantissant que le secret pourra être reconstruit si nécessaire.

Nous avons ensuite présenté un schéma d'engagement, permettant de masquer des valeurs et de les révéler ensuite. Le schéma présenté est notamment utilisé pour calculer des preuves NIZK [GS08], qui permettent de prouver des équations dans les groupes bilinéaires sans révéler certaines valeurs secrètes.

Finalement, nous avons décrit trois schémas de signature : un premier, classique et efficace, qui n'est cependant pas compatible avec le système de preuves de Groth et Sahai ; un schéma automorphe, que l'on peut combiner avec ces preuves pour construire des signatures proxy anonymes, qui permettent de déléguer la capacité de signer des messages en révélant uniquement le premier utilisateur de la chaîne de délégation.

3 État de l'art

Nous présentons dans ce chapitre l'état de l'art des mécanismes de réputation préservant la vie privée, en utilisant la terminologie présentée au chapitre 1. Nous commençons par expliquer pourquoi les mécanismes de réputation permettent d'instaurer de la confiance entre des utilisateurs inconnus et comment ces mécanismes sont utiles lors d'interactions en ligne. Nous présentons également les caractéristiques qu'ils doivent présenter pour être utiles.

Nous nous intéressons ensuite aux moteurs de réputation, c'est-à-dire aux différentes manières existantes de calculer les scores de réputation. Nous extrayons de cette présentation les qualités nécessaires aux moteurs de réputation – dans la suite, nous supposons que le moteur de réputation utilisé possède ces qualités.

Nous abordons par la suite les attaques existantes sur les mécanismes de réputation, ainsi que les contre-mesures efficaces. Nous expliquons en outre que garantir les propriétés de vie privée et de sécurité définies en section 1.4 protègent un mécanisme de réputation de la plupart de ces attaques. Parmi les contre-mesures à ces attaques, préserver la vie privée des fournisseurs de service et des clients permet d'empêcher aussi bien les médisances des clients que les discriminations des fournisseurs. Il est également nécessaire de préserver la vie privée des utilisateurs pour se conformer au cadre légal. Nous expliquons les conséquences qu'implique la directive européenne à propos de la protection des données [Eur95] sur les mécanismes de réputation, et détaillons les propriétés de vie privée que les mécanismes de réputation doivent garantir à leurs utilisateurs.

Nous sommes alors capables de présenter les mécanismes de réputation préservant la vie privée existants. Les premiers mécanismes se contentent de préserver le secret des notes. Ensuite, d'autres mécanismes intègrent le respect de la vie privée des utilisateurs au cœur de leur conception. Nous analysons tous ces mécanismes sous l'angle des propriétés de sécurité et de vie privée définies en section 1.4.

Finalement, nous dressons le bilan de cet état de l'art, et expliquons que la conception d'un mécanisme de réputation utile et préservant la vie privée de ses utilisateurs requiert de préserver l'anonymat des fournisseurs tout en permettant aux clients d'émettre des témoignages positifs et négatifs de manière indéniable. Jusqu'à présent, aucun mécanisme ne parvient à combiner ces deux propriétés ; notre proposition est la seule à le faire.

3.1 Réputation et confiance

Depuis quelques années, les mécanismes de réputation ont permis à des plateformes telles que Airbnb [1], BlaBlaCar [2] ou Stack Overflow [3] de prendre leur essor. Ces plateformes permettent

1. https://www.airbnb.com
2. https://www.blablacar.com
3. https://www.stackoverflow.com

respectivement de louer des appartements, de proposer des covoiturages ou de répondre à des questions techniques. Leur point commun est l'intégration d'un mécanisme de réputation. Ces mécanismes de réputation permettent à un client potentiel de se construire une opinion sur les fournisseurs, jusqu'alors inconnus, et de choisir s'il est raisonnable de leur faire *confiance*. Ainsi, les mécanismes de réputation sont des outils dont l'objectif principal est de créer de la confiance entre des utilisateurs ne se connaissant pas. Dans cette section, nous définissons la confiance et étudions les qualités nécessaires aux mécanismes de réputation pour inciter des utilisateurs ne se connaissant pas à se faire confiance.

Afin de déterminer les qualités essentielles au bon fonctionnement d'un mécanisme de réputation, il faut dans un premier temps s'intéresser aux caractéristiques de la confiance. De manière générale, Gambetta propose de définir la confiance comme suit :

> Trust (or, symmetrically, distrust) is a particular level of the subjective probability with which an agent assesses that another agent or group of agents will perform a particular action, both *before* he can monitor such action (or independently of his capacity ever to be able to monitor it) *and* in a context in which it affects *his own* action. (Diego Gambetta [Gam90])

Cette définition possède quatre points remarquables :

1. Tout d'abord, la confiance est une *probabilité*, c'est-à-dire qu'un client ne peut jamais être certain du comportement d'un fournisseur.
2. Ensuite, cette probabilité est *subjective* : deux clients ne placent pas nécessairement la même confiance en un fournisseur.
3. La confiance est également *contextuelle*, et s'applique à une action particulière attendue du fournisseur.
4. Finalement, la confiance dépend des attentes du client : quelles sont les conséquences dans le cas où le fournisseur se comporte mal ? S'il est crucial que le fournisseur se comporte correctement, alors il sera plus difficile pour le client de lui faire confiance.

Resnick et Zeckhauser expliquent qu'il est facile d'avoir confiance dans un magasin ayant pignon sur rue grâce aux nombreux indices à disposition [RZ02] : (a) il est possible d'inspecter les biens vendus dans une enseigne pour vérifier leur qualité ; (b) les interactions avec la boulangerie du quartier sont fréquentes, et il est donc facile de connaître la qualité de leur pain ; (c) les amis, voisins et collègues nous recommandent les magasins qu'ils ont appréciés ou, au contraire, nous déconseillent ceux qui les ont déçus ; (d) les vendeurs peuvent être rencontrés lors d'activités sociales : à l'école, dans une salle de sport, … ; (e) le nom d'une chaîne de magasins donne une première information sur la qualité des produits vendus ; (f) l'emplacement d'un magasin peut également en dire beaucoup : un magasin sur les Champs-Élysées indique des revenus suffisants, et donc une clientèle fournie. Cependant, aucun de ces signes indicateurs n'est présent lors d'un achat en ligne, créant ainsi une *asymétrie d'information* entre le fournisseur de service, connaissant la valeur des biens vendus, et le client, ignorant.

Dans *The Market for "Lemons"*, Akerlof montre les effets d'une telle asymétrie d'information à travers le marché des voitures d'occasion [Ake70]. Deux types de voiture sont vendues : des bonnes occasions, correctement entretenues, ainsi que des occasions en mauvais état – « lemon » est un terme d'argot désignant une guimbarde. Les vendeurs de bonnes occasions

espèrent tirer un bon prix de leur voiture, ainsi leur prix est supérieur à celui des guimbardes. Au moment où un acheteur s'intéresse à une voiture, il ne sait pas dans laquelle des deux catégories elle se place. Comme un acheteur risque de tomber sur une voiture en mauvais état, il n'a pas envie de payer cher et d'obtenir une guimbarde. Finalement, seules les voitures vendues à bas prix trouvent acquéreur, c'est-à-dire les mauvaises occasions : les propriétaires de bonnes occasions ne désirent pas vendre leur voiture en dessous d'un certain prix, trop élevé pour un acheteur qui craint de récupérer une guimbarde. Ce phénomène devrait également s'appliquer aux plateformes en ligne : les clients ne connaissent pas les fournisseurs, et n'ont aucun moyen de s'assurer de la qualité des services fournis.

C'est pour réduire cette asymétrie d'information que les plateformes en ligne intègrent des mécanismes de réputation. La réputation est définie par le Littré comme « l'opinion que le public a d'une personne » [Lit74]. Les mécanismes de réputation utilisent ainsi les opinions des témoins d'un fournisseur, c'est-à-dire leurs témoignages, pour calculer le score de réputation de ce fournisseur. Resnick et Zeckhauser s'intéressent plus particulièrement à la plateforme d'enchères électroniques eBay [4]. Dans cette plateforme, un fournisseur propose un objet à la vente en en fournissant une description. Les clients potentiels enchérissent afin de remporter l'objet. Une fois que l'enchère est finie et la transaction effectuée, le client et le fournisseur peuvent chacun noter l'autre afin de refléter le comportement de leur partenaire. Une note peut être soit négative (−1), soit neutre (0), soit positive (+1). Ces notes sont finalement agrégées pour obtenir les scores de réputation des utilisateurs : la réputation d'un fournisseur de service est la somme des notes qu'il a reçues. Resnick et Zeckhauser étudient ce mécanisme de réputation [RZ02], et s'intéressent plus précisément au taux de notes positives, en faisant l'hypothèse que plus ce taux est élevé, plus les utilisateurs sont satisfaits par la plateforme. En analysant les transactions effectuées entre février et juin 1999, ils trouvent que plus de 99% des notes émises sont positives ! Ainsi, une écrasante majorité des clients et des fournisseurs sont satisfaits par leurs transactions. Resnick et Zeckhauser expliquent l'efficacité des mécanismes de réputation de la manière suivante : les fournisseurs pensent qu'en se comportant mal, ils vont recevoir des témoignages négatifs ; les clients fondent leurs opinions sur la réputation des fournisseurs ; ainsi, les fournisseurs ont intérêt à bien se comporter pour obtenir une excellente réputation, et donc attirer de nouveaux clients. Les mécanismes de réputation sont également intéressant pour les fournisseurs ; en effet, en étudiant quatorze travaux empiriques et expérimentaux sur les mécanismes de réputation, Dellarocas montre qu'un fournisseur ayant une réputation élevée peut vendre un service plus cher qu'un fournisseur de moindre réputation [Del06]. Ainsi, avoir une bonne réputation est un avantage économique pour les fournisseurs de service, qui compense les efforts effectués afin de fournir un service correct. Dans le premier chapitre de sa thèse, Ravoaja [Rav08] se penche sur les liens entre réputation, confiance et coopération, en particulier pour les systèmes pair à pair ; il explique plus précisément pourquoi la réputation incite à faire confiance, et crée ainsi des coopérations.

Finalement, Resnick et coll. [Res+00] présentent les trois propriétés nécessaires pour qu'un mécanisme de réputation puisse fonctionner :

1. les fournisseurs de service doivent vivre à long terme, et espérer continuer à interagir dans le futur ;

4. https://www.ebay.com

2. les témoignages doivent refléter les interactions courantes, et être visibles dans le futur ;
3. les témoignages doivent guider la confiance des clients.

Un mécanisme de réputation garantissant ces trois propriétés permet d'instaurer de la confiance entre des utilisateurs ne se connaissant pas : puisque les fournisseurs de service espèrent continuer à vendre leurs services, ils doivent se comporter convenablement afin d'obtenir des témoignages positifs, et attirer ainsi de nouveaux clients.

En plus de ces trois critères, Baumeister et coll. [Bau+01] expliquent que « Bad feedback has stronger effects than good feedback », c'est-à-dire que les interactions qui se déroulent mal ont plus d'influence sur nos opinions que les interactions se déroulant sans accrocs :

> When people first learn about one another, bad information has a significantly stronger impact on the total impression than any comparable good information.

Les mécanismes de réputation, visant à imiter les opinions que nous formons dans le monde réel, doivent donc prendre les mauvais témoignages en compte. En effet, un fournisseur de service s'étant comporté cent fois correctement et une seule fois incorrectement peut être considéré de confiance, tandis qu'un fournisseur de service s'étant comporté cent fois correctement et mille fois incorrectement ne l'est clairement pas. Cependant, sans témoignages négatifs, ces deux fournisseurs sont indistinguables. Nous précisons donc le deuxième critère en ajoutant que les témoignages doivent pouvoir refléter les interactions qui se sont correctement déroulées, ainsi que celles qui se sont incorrectement déroulées. Comme nous le verrons au cours des chapitres 4 et 5, nos mécanismes de réputation garantissent ces propriétés.

3.2 Moteurs de confiance et de réputation

Maintenant que nous avons vu quels sont les prérequis des mécanismes de réputation, nous nous intéressons à la manière dont les mécanismes existants calculent les scores de réputation, c'est-à-dire que nous nous intéressons à leurs moteurs de réputation.

3.2.1 Moteurs de réputation discrets

Abdul-Rahman et Hailes proposent un moteur de confiance discret [AH00]. À l'issue d'une transaction, le client peut considérer quatre valeurs pour le comportement du fournisseur : très bon, bon, mauvais, très mauvais. Le client peut ensuite combiner ses propres témoignages pour obtenir une confiance, qui est alors dite *directe*, et peut elle aussi prendre quatre valeurs : le fournisseur est très digne de confiance, digne de confiance, peu digne de confiance ou absolument pas digne de confiance. La confiance directe qu'un client place en un fournisseur est définie de la manière suivante. Soient n_{tb}, n_{b}, n_{m}, n_{tm} le nombre d'observations directes du comportement de ce fournisseur. Si $(n_{\text{tb}}, n_{\text{b}}, n_{\text{m}}, n_{\text{tm}})$ admet un unique maximum, alors la confiance correspond à ce maximum – par exemple, si le maximum est n_{b}, alors le client juge le fournisseur digne de confiance. En plus de cette confiance directe, les auteurs proposent une méthode permettant d'agréger les témoignages émis par d'autres clients du fournisseur, en prenant également en compte la confiance dans ces clients.

Cependant, Jøsang et coll. [JIB07] notent que l'emploi de mesures discrètes impliquent des heuristiques de calcul peu fiables. Par exemple, considérons un client ayant observé (n_{tb} =

$10, n_b = 0, n_m = 9, n_{tm} = 9$) à propos d'un fournisseur donné ; d'après les heuristiques d'Abdul-Rahman et de Hailes, ce client juge ce fournisseur très digne de confiance alors que ce fournisseur a déçu ses clients passés dix-huit fois sur vingt-huit interactions.

3.2.2 Systèmes bayésiens

Jøsang et Ismail proposent de modéliser le comportement d'un fournisseur par un système bayésien, en utilisant une loi de probabilité bêta [JI02]. Une telle loi est paramétrée par deux paramètres de forme, a et b, que les clients cherchent à estimer pour obtenir le score de réputation d'un fournisseur. L'intuition derrière cette idée est simple : un fournisseur de service réputé est un fournisseur se comportant correctement dans la majorité des cas, mais à qui il peut ponctuellement arriver des mésaventures. Afin de représenter simplement les scores de réputation, les auteurs proposent de s'intéresser plus particulièrement à l'espérance de cette loi bêta. Ainsi, un fournisseur de service ayant reçu r notes positives et s notes négatives a la réputation suivante :

$$\text{rep} = \mathbb{E}\Big(\text{beta}(p|a = r+1, b = s+1)\Big) = \frac{r+1}{r+s+2} \in [0,1].$$

De cette façon, pour un fournisseur ayant reçu 8 notes positives et 2 notes négatives, on obtient rep = 0,75.

Les auteurs proposent également de diminuer l'importance des témoignages en fonction du temps écoulé depuis leur émission ; en effet, un témoignage émis trois mois plus tôt n'est plus nécessairement pertinent. À cet effet, ils introduisent un facteur de vieillissement $\lambda \in [0,1]$. Si $\Big((r_1,s_1),\ldots,(r_n,s_n)\Big)$ est une séquence de notes données sur un fournisseur, respectivement émises aux temps $t_1,\ldots,t_n$, alors les combinaisons de ces notes sont :

$$r(t) = \sum_{i=1}^{n} r_i \cdot \lambda^{t-t_i} \qquad \text{et} \qquad s(t) = \sum_{i=1}^{n} s_i \cdot \lambda^{t-t_i}.$$

Pour $\lambda = 1$, tous les témoignages ont la même importance, tandis que pour $\lambda = 0$, seuls les témoignages émis à l'instant t sont pris en compte.

Les auteurs expliquent que la valeur des transactions est importante ; en effet, le comportement d'un fournisseur lors d'une transaction d'une valeur de 100 € est plus significatif que lors d'une transaction d'une valeur de 10 €. Ainsi, un témoignage sur une transaction d'une valeur de 100 € est plus important qu'un autre témoignage sur une transaction d'une valeur de 10 €.

Finalement, en notant $(r_1,s_1),\ldots,(r_n,s_n)$ les témoignages reçus par un fournisseur donné aux temps $t_1,\ldots,t_n$, et $v_1,\ldots,v_n$ la valeur des transactions associées, la réputation de ce fournisseur au temps t pour un facteur de vieillissement λ est définie par :

$$\text{rep}_\lambda(t) = \frac{1 + \sum_{i=1}^{n} r_i \cdot v_i \cdot \lambda^{t-t_i}}{2 + \sum_{i=1}^{n} (r_i + s_i) \cdot v_i \cdot \lambda^{t-t_i}}.$$

Cette méthode peut être simplifiée en normalisant les r_i et s_i de manière à ce que $r'_i + s'_i = 1$. De cette manière, un témoin ne peut pas amplifier l'importance de son témoignage en choisissant

par exemple $r_i = 0$ et $s_i = 10$; le poids d'un témoignage est uniquement lié à son importance et à son âge. Alors, la réputation devient :

$$\mathrm{rep}_\lambda(t) = \frac{1 + \sum_{i=1}^{n} r'_i \cdot v_i \cdot \lambda^{t-t_i}}{2 + \sum_{i=1}^{n} v_i \cdot \lambda^{t-t_i}}.$$

L'avantage de cette méthode est que le score de réputation final d'un fournisseur de service est une valeur comprise entre 0 et 1 : plus la réputation est proche de 1, plus le fournisseur est digne de confiance ; il est donc facile de comprendre une telle réputation et de comparer deux fournisseurs : le choix des clients est facilité.

3.2.3 Théorie de Dempster-Shafer

Jøsang et Ismail [JI02] proposent également une méthode ayant deux avantages principaux : les témoins peuvent faire part de leur incertitude, et les utilisateurs peuvent utiliser la confiance qu'ils placent en un témoin pour pondérer ses témoignages. À cet effet, ils utilisent la méthode de Jøsang [Jøs99] qui repose sur la théorie de Dempster-Shafer [Sha76] ; cette méthode représente la note d'un client par un quadruplet (b,d,u,a) où $b+d+u=1$ et $b,d,u \in [0,1]$; b reflète la conviction (*belief*), d la méfiance (*disbelief*), u l'incertitude (*uncertainty*) et a l'atomicité relative, déterminant à quel degré l'incertitude participe au score de réputation. Ce moteur définit la réputation comme

$$\mathrm{rep} = b + a \cdot u.$$

Par exemple, une note de $(1;0;0;*)$ indique que le client est convaincu que le fournisseur s'est correctement comporté et donne $\mathrm{rep} = 1$, tandis qu'une note de $(0{,}4;0{,}1;0{,}5;0{,}5)$ indique que le client pense que le fournisseur s'est bien comporté sans en être convaincu, et donne $\mathrm{rep} = 0{,}65$.

Jøsang propose également de combiner les témoignages de deux manières. La première, appelée *recommandation*, permet la transitivité de la confiance : si X fait confiance à Y et si Y fait confiance à Z, alors X pourra également faire confiance à Z. La deuxième manière, appelée *consensus*, permet de combiner deux opinions ; par exemple, si X fait confiance à Y_1 et à Y_2, et que ces deux utilisateurs font confiance à Z, le consensus permet de calculer la confiance que X porte en Z à travers les deux chemins XY_1Z et XY_2Z. Ainsi, ces deux opérations permettent de construire un graphe orienté à partir de la confiance, où les nœuds sont les utilisateurs et les arêtes les relations de confiance.

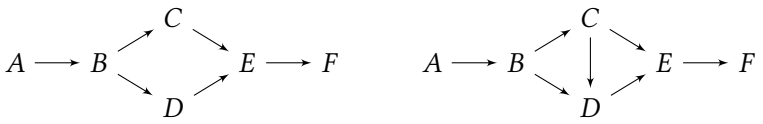
Plus formellement, Jøsang et Ismail définissent ces opérations à partir des opérations classiques de la théorie de Dempster-Shafer [Sha76]. Considérons trois agents X, Y et Z. Si $(b_{X\to Y}, d_{X\to Y}, u_{X\to Y}, a_{X\to Y})$ représente la confiance de X dans les recommandations de Y et $(b_{Y\to Z}, d_{Y\to Z}, u_{Y\to Z}, a_{Y\to Z})$ la confiance de Y en Z, alors la confiance de X en Z à travers la recommandation de Y est donnée par :

$$\begin{cases} b_{X\overset{Y}{\to}Z} = b_{X\to Y} \cdot b_{Y\to Z} \\ d_{X\overset{Y}{\to}Z} = b_{X\to Y} \cdot d_{Y\to Z} \\ u_{X\overset{Y}{\to}Z} = d_{X\to Y} + u_{X\to Y} + b_{X\to Y} \cdot u_{Y\to Z} \\ a_{X\overset{Y}{\to}Z} = a_{Y\to Z}. \end{cases}$$

L'opération de consensus entre deux opinions $X \to Z$ et $Y \to Z$ donne l'opinion $(X, Y) \to Z$ définie par :

$$\begin{cases} b_{X,Y\to Z} = (b_{X\to Z} \cdot u_{Y\to Z} + b_{Y\to Z} \cdot u_{X\to Z})/\kappa \\ d_{X,Y\to Z} = (d_{X\to Z} \cdot u_{Y\to Z} + d_{Y\to Z} \cdot u_{X\to Z})/\kappa \\ u_{X,Y\to Z} = (u_{X\to Z} \cdot u_{Y\to Z})/\kappa \\ a_{X,Y\to Z} = \dfrac{a_{Y\to Z} \cdot u_{X\to Z} + a_{X\to Z} \cdot u_{Y\to Z} - (a_{X\to Z} + a_{Y\to Z}) \cdot u_{X\to Z} \cdot u_{Y\to Z}}{u_{X\to Z} + u_{Y\to Z} - 2 \cdot u_{X\to Z} \cdot u_{Y\to Z}}, \end{cases}$$

où $\kappa = u_{X\to Z} + u_{Y\to Z} - u_{X\to Z} \cdot u_{Y\to Z}$ pour $(u_{X\to Z}, u_{Y\to Z}) \notin \{(0,0),(1,1)\}$. Cependant, Jøsang explique qu'un consensus ne peut être effectué que si les témoignages combinés sont indépendants, c'est-à-dire que le témoignage d'un utilisateur doit être pris en compte une seule fois. Pour comprendre ce problème, considérons les graphes de confiance présentés en figure 3.1, où les lettres A, B, C, D, E et F représentent les utilisateurs, et les flèches représentent les relations de confiance ; par exemple, $A \to B$ signifie que A fait confiance à B.



(a) Graphe de confiance calculable (b) Graphe de confiance incalculable

FIGURE 3.1 – Combinaisons de recommandation et de consensus [Jøs99]

Dans le cas du premier graphe, il y a deux chemins possibles allant de A à F : $ABCEF$ et $ABDEF$, qui peuvent être transformés en un seul chemin $AB(C,D)EF$; à ce moment là, les seuls témoignages combinés sont BCE et BDE, qui sont bien indépendants. Cependant, dans le cas du deuxième graphe, le chemin de C à D pose problème. En effet, il faut maintenant faire un consensus de trois chemins – BCE, BDE et $BCDE$ –, qui ne sont pas indépendants : l'arête BC est commune à BCE et à $BCDE$, tandis que l'arête DE est commune à BDE et à $BCDE$. Ainsi, il faut faire un choix pour calculer la réputation : ignorer BDE ou $BCDE$ et combiner les témoignages restants – ce qui n'utilise pas tous les témoignages disponibles –, ou combiner les témoignages en dépit de leur dépendance. Dans les deux cas, le calcul de la réputation est incorrect.

Yu et Singh [YS02] proposent une approche similaire, reposant également sur la théorie de Dempster-Shafer. Dans leur cas, les témoignages sont automatiquement collectés grâce à une architecture distribuée : chaque utilisateur maintient des liens avec ses voisins dans le système. Lorsqu'un client cherche à estimer la confiance qu'il peut avoir en un fournisseur, il y a deux cas : (a) soit le client a déjà interagi avec ce fournisseur, et le connaît donc assez bien ; (b) soit le client n'a pas assez interagi avec ce fournisseur. Dans le premier cas, le client utilise uniquement les témoignages qu'il a émis sur le fournisseur pour estimer sa confiance. Dans le second cas, le client fait appel à ses voisins pour obtenir plus d'informations sur le fournisseur. Ses voisins lui font alors part de leurs témoignages. Le client est finalement capable d'estimer sa confiance à partir de tous les témoignages reçus.

Pour résumer, la méthode de Dempster-Shafer permet une prise en compte fine des témoi-

gnages en fonction de la confiance dans les témoins, tout en permettant aux témoins de faire part de leur incertitude. Cependant, les méthodes de calcul sont complexes et, dans certains cas, ne permettent pas de calculer correctement le score de réputation des fournisseurs dans tous les cas.

3.2.4 Moteurs reposants sur des architectures pair à pair

Kamvar et coll. proposent Eigentrust [KSG03], un mécanisme de réputation reposant sur une architecture distribuée afin de prendre cinq considérations en compte :

1. L'autogestion du système : une autorité centrale ne doit pas être nécessaire pour faire respecter les règles du système.
2. L'anonymat des fournisseurs : ils ne doivent pas être connus par un identifiant les représentant, comme leur adresse IP, mais par un identifiant opaque, par exemple une séquence aléatoire de caractères. Notons que cette notion d'anonymat n'a rien à voir avec notre définition de vie privée des fournisseurs (propriété 1) : l'identifiant des fournisseurs, même opaque, reste unique et permet de tracer leurs transactions.
3. Les nouveaux arrivants ne doivent pas être favorisés, sous peine de voir les fournisseurs ayant une mauvaise réputation se réinscrire pour repartir de zéro.
4. Pas de surcouts : ce mécanisme ne doit pas nécessiter une puissance calculatoire ou une capacité de stockage particulière.
5. Le mécanisme doit résister aux attaques collectives des utilisateurs.

Comme ce mécanisme ne peut pas se reposer sur une autorité centrale, la question du stockage des témoignages et des scores de réputation se pose. Les auteurs expliquent qu'un fournisseur ne doit pas stocker ses propres témoignages ; en effet, il pourrait alors facilement les manipuler. C'est pourquoi Kamvar et coll. proposent d'assigner un nombre prédéfini de *gestionnaires de score* à chaque fournisseur, choisis aléatoirement. Au moment où un client veut connaître la réputation d'un fournisseur, il peut interroger les gestionnaires de score pour l'obtenir. Comme un nombre suffisamment grand de gestionnaires de score est assigné à chaque fournisseur, la probabilité qu'une majorité d'entre eux soit malveillante et parvienne à modifier les témoignages est relativement faible.

Les auteurs proposent d'utiliser une table de hachage distribuée (DHT) pour gérer les témoignages, comme CAN [Rat+01] ou Chord [Sto+03]. Le principe de cet outil est de placer les utilisateurs aléatoirement sur un espace – un espace bidimensionnel dans le cas de CAN et un anneau pour Chord – en utilisant une fonction de hachage sur un identifiant, par exemple l'adresse IP et le port TCP utilisés. Chaque utilisateur est ensuite chargé de maintenir un espace particulier. Ainsi, dans le cas de Chord, les n utilisateurs les plus proches d'un fournisseur sur l'anneau sont ses gestionnaires de score.

Similairement aux moteurs de réputation utilisant la théorie de Dempster-Shafer, EigenTrust permet de prendre en compte la réputation des témoins pour calculer la réputation d'un fournisseur. Chaque témoin i du fournisseur j a une valeur de confiance locale s_{ij}, construite à partir des transactions passées. Ces valeurs sont normalisées pour faire en sorte que la somme

des valeurs de confiance d'un utilisateur donne 1, afin que chaque témoin contribue autant au calcul du score de réputation :

$$c_{ij} = \frac{\max(s_{ij}, 0)}{\sum_j \max(s_{ij}, 0)}.$$

Considérons maintenant un client i, un fournisseur de service k, et les n témoins de k. Si la valeur de confiance normalisée de i en le témoin j est c_{ij}, les valeurs de confiance locales des témoins sont agrégées par :

$$t_{ik} = \sum_j c_{ij} \cdot c_{jk},$$

où t_{ik} représente la confiance que i porte en k à travers ses connaissances. Cette équation peut également se mettre sous forme matricielle. Elle donne alors $\vec{t_i} = C^\intercal \vec{c_i}$, c'est-à-dire :

$$\begin{pmatrix} t_{i1} \\ \vdots \\ t_{in} \end{pmatrix} = \begin{pmatrix} c_{11} & \cdots & c_{n1} \\ \vdots & & \vdots \\ c_{1n} & \cdots & c_{nn} \end{pmatrix} \begin{pmatrix} c_{i1} \\ \vdots \\ c_{in} \end{pmatrix}.$$

Notons que de cette manière, seuls les chemins de longueur 1 sont pris en compte : il y a à chaque fois un seul intermédiaire entre i et j. Afin d'augmenter la longueur de ces chemins, il suffit de continuer à multiplier par $C^\intercal$ à gauche. Ainsi, $\vec{t_i} = (C^\intercal)^n \vec{c_i}$ prend en compte tous les chemins de longueur inférieure ou égale à n. En allant plus loin, les auteurs prouvent que cette confiance converge et que, quel que soit le client i cherchant à la connaître, il obtient la même valeur, qui se trouve être le vecteur propre principal de la matrice C – d'où le nom EigenTrust.

Kamvar et coll. proposent également une méthode permettant de calculer cette confiance de manière distribuée tout en évitant qu'une collusion d'utilisateurs ne perturbe le résultat.

Anceaume et Ravoaja proposent également un moteur de réputation pour une architecture distribuée [AR06]. Similairement au mécanisme précédent, les auteurs proposent de collecter les témoignages en créant des chaînes de recommandation. Cependant, ils n'utilisent pas la réputation des utilisateurs pour pondérer leurs témoignages ; à la place, chaque utilisateur maintient une *crédibilité* pour chaque témoin lui ayant fourni une recommandation. Lorsqu'une recommandation est pertinente, la crédibilité d'un témoin est améliorée, et elle est réciproquement diminuée lorsque le comportement du fournisseur diffère de la recommandation. Plus précisément, notons p le client désirant connaître la réputation du fournisseur de service s. Dans un premier temps, p demande à chaque témoin de l'ensemble $\mathcal{P}_p^s(t)$ des témoins du fournisseur s de lui envoyer ses témoignages issus de transactions récentes. Il obtient alors $\left\{\left(\mathrm{obs}_k^s(t_0), t_0\right), \ldots, \left(\mathrm{obs}_k^s(t_\ell), t_\ell\right)\right\}$ pour chaque témoin k. Le client calcule ensuite le nombre minimum de témoignages envoyés par un témoin, qu'il note f. Pour chaque témoin, p ne conserve que les f derniers témoignages ; l'ensemble de ces f témoignages est noté $\mathcal{F}_k^s(t)$. De cette manière, chaque témoin apporte exactement f témoignages. Finalement, la réputation du fournisseur s est calculée de la manière suivante :

$$\mathrm{rep}_p^s(t) = \frac{1}{\sum\limits_{k \in \mathcal{P}_p^s(t)} c_{p,k}^s(t)} \cdot \sum_{k \in \mathcal{P}_p^s(t)} c_{p,k}^s(t) \rho_k(t),$$

où $c^s_{p,k}(t)$ représente la crédibilité de k pour témoigner sur s à l'instant t vue par p, et $\rho^s_k(t)$ est défini par :

$$\rho^s_k(t) = \frac{1}{f} \cdot \sum_{\left(\mathrm{obs}^s_k(t'),t'\right) \in \mathcal{F}^s_k(t)} \mathrm{obs}^s_k(t').$$

La crédibilité de k vue par p est initialement définie par une valeur par défaut c_0. Ensuite, lors de chaque recommandation de k sur le fournisseur s pour p, cette crédibilité devient :

$$1 - |\rho^s_k(t) - \rho^s_p(t)|^\alpha,$$

où $\alpha > 0$ est un paramètre définissant la précision nécessaire aux recommandations.

Ravoaja et Anceaume proposent en outre une architecture décentralisée permettant le stockage des témoignages [RA07]. À l'instar d'EigenTrust, cette architecture repose sur une DHT, à savoir Chord [Sto+03]. Plus concrètement, les fournisseurs de service sont organisés sur une première DHT. Pour chaque fournisseur, une deuxième DHT permet d'organiser les témoins. Les témoins eux-mêmes s'insèrent dans celle-ci, et donnent leurs témoignages aux clients potentiels. Ainsi, n'importe qui peut recommander un fournisseur ; cependant, la crédibilité écarte rapidement les témoins mentant à propos du comportement du fournisseur.

Marti et Garcia-Molina proposent une taxonomie des mécanismes de réputation fonctionnant à partir d'un réseau pair à pair [MG06], et classent les mécanismes existants en fonction par exemple de la manière dont les témoignages sont obtenus et agrégés, de la sélection des témoins, ou du format des témoignages. De manière générale, de tels mécanismes de réputation proposent des algorithmes de calcul de réputation passant à l'échelle.

3.2.5 Modèle de Markov caché

Elsalamouny et Sassone [ES13] expliquent que la plupart des moteurs de réputation utilisent la bêta réputation de Jøsang et Ismail [JI02]. Cependant, et malgré le principe de vieillissement des témoignages utilisé, de tels moteurs de réputation ne permettent pas de modéliser les comportements *dynamiques*. Considérons un fournisseur qui se comporte bien pendant une longue période de temps, puis se comporte mal pendant une courte période de temps avant de recommencer à bien se comporter. Sa réputation ne sera pas précise pendant le court moment où il se comporte mal, ce qui est problématique.

Pour surmonter ces difficultés, Elsalamouny et Sassone proposent de modéliser le comportement des fournisseurs de service par un modèle de Markov caché (HMM) [ES13]. Un HMM est un quintuplet, qui comporte les éléments suivants :

- $S = \{s_1, \ldots, s_n\}$, un ensemble d'états, cachés.[4]
- $V = \{v_1, \ldots, v_k\}$, un ensemble d'observables.[4]
- π, une distribution sur S, représentant les états initiaux ;
- $A = (a_{ij})_{1 \leqslant i,j \leqslant n}$, la matrice de transition d'états, telle que $\forall i, \sum_j a_{i,j} = 1$; a_{ij} représente la probabilité de transition de l'état s_i à l'état s_j.

4. Les ensembles S et V peuvent être infinis, et même indénombrables ; cependant, les considérer finis permet de simplifier la compréhension des HMM.

— $B = (b_{ij})_{\substack{1 \leqslant i \leqslant n \\ 1 \leqslant j \leqslant k}}$, la matrice d'émission, telle que $\forall i, \sum_j b_{ij} = 1$; b_{ij} représente la probabilité que l'état s_i donne l'observable v_j.

L'ensemble S représente les différentes tendances que le comportement du fournisseur de service peut suivre. Par exemple, si le fournisseur suit deux tendances, $S = \{s_1, s_2\}$. L'ensemble V représente les comportements du fournisseur. Si l'on considère que le fournisseur peut bien ou mal se comporter, alors $V = \{v_1 = \text{bon}, v_2 = \text{mauvais}\}$; l'ensemble V peut également être continu ($V = [0,1]$). π est la distribution des états initiaux ; si les deux états initiaux sont équiprobables, alors $\pi(s_1) = \pi(s_2) = 0{,}5$. La matrice A représente les transitions possibles entre les états. L'exemple où le fournisseur se comporte bien sur de longues périodes et mal sur de courtes périodes pourrait donner

$$A = \begin{bmatrix} 0{,}9 & 0{,}1 \\ 0{,}4 & 0{,}6 \end{bmatrix},$$

ce qui signifie que lorsque le fournisseur est dans l'état s_1, la probabilité qu'il reste dans s_1 est de 0,9, tandis que la probabilité qu'il aille dans s_2 est de 0,1. Finalement, la matrice B représente le comportement du fournisseur en fonction de son état actuel. Par exemple,

$$B = \begin{bmatrix} 0{,}95 & 0{,}05 \\ 0{,}1 & 0{,}9 \end{bmatrix}$$

signifie que que lorsque le fournisseur est dans l'état s_1, la probabilité qu'il se comporte bien est de 0,95. En revanche, s'il est dans l'état s_2, la probabilité qu'il se comporte mal est de 0,9. Notons que la bêta réputation est un HMM pour $n = 1$, c'est-à-dire pour un seul état.

Elsalamouny et Sassone proposent d'estimer le HMM d'un fournisseur de service en utilisant l'algorithme de Baum-Welch [Bau+70]. Cet algorithme permet d'estimer π, A et B à partir d'une séquence de résultats, en ayant fixé préalablement le nombre d'états et de comportements observés. Ils proposent en outre d'utiliser les témoins pour améliorer cette estimation, tout en prenant leur réputation en compte. Finalement, Elsalamouny et Sassone montrent expérimentalement que, dans le cas d'un fournisseur exhibant un comportement dynamique, leur méthode est plus précise que la bêta réputation avec vieillissement des témoignages.

Cette méthode de calcul des scores de réputation est donc très précise, et permet de modéliser des comportements de fournisseur évolués. Cependant, le score de réputation d'un fournisseur de service est un HMM, c'est-à-dire un quintuplet (S, V, π, A, B) ; il est plus difficile de comparer deux HMM pour savoir quel fournisseur présente moins de risques. Au contraire, la bêta réputation, bien que moins raffinée, permet de résumer très succinctement le comportement d'un fournisseur de service.

3.2.6 Bilan

Nous avons vu cinq principaux types de moteurs de réputation. Les premiers, reposant sur une discrétisation des scores de réputation, sont très faciles à utiliser et à comprendre pour les utilisateurs. Cependant, les heuristiques de calcul sont fallacieuses et peuvent donner des scores de réputation incohérents. La bêta réputation utilise des systèmes bayésiens pour donner une estimation du comportement des fournisseurs résumée par un simple nombre, tout en prenant

en compte l'âge des témoignages ou l'importance des transactions ; cependant, cette méthode peut être limitée dans certains cas, notamment lorsque le comportement des fournisseurs est dynamique. Nous avons ensuite présenté les méthodes utilisant la théorie de Dempster-Shafer, permettant de prendre aussi bien l'incertitude que la confiance dans les témoins en compte. Cette méthode de calcul est saine mais les scores de réputation ne sont pas toujours calculables, ce qui est problématique ; de plus, cette méthode demande de reconstruire le graphe de confiance, ce qui pose des problèmes de vie privée. Nous avons également décrit des méthodes reposant sur des architectures pair à pair, qui passent à l'échelle tout en prenant en compte la confiance dans les témoins, mais peuvent demander de connaître le graphe de confiance. Finalement, nous avons vu une méthode utilisant des HMM ; cette méthode est la seule à traiter correctement les comportements dynamiques des fournisseurs, mais est très complexe et la réputation d'un fournisseur ne peut être simplement résumée. La table 3.1 récapitule les avantages et inconvénients de chacun de ces moteurs.

Moteur	Avantages	Inconvénients
Discret	interprétation facile	heuristiques peu fiables
Bayésien	interprétation facile âges et valeurs	comportements dynamiques
Dempster-Shafer	interprétation facile confiance dans les témoins	graphe de confiance parfois incalculable
P2P	passage à l'échelle interprétation facile confiance dans les témoins	graphe de confiance
HMM	comportements dynamiques âges et valeurs confiance dans les témoins	interprétation complexe

TABLE 3.1 – Avantages et inconvénients des moteurs de réputation

Jøsang et coll. [JIB07] proposent une vue d'ensemble des mécanismes de réputation existants, et concluent qu'un moteur de réputation doit garantir quatre propriétés :

1. Les scores de réputation doivent, à long terme, représenter correctement le comportement des fournisseurs de service ;
2. Les témoignages récents sur le comportement d'un fournisseur doivent avoir plus de poids que les anciens ;
3. Les moteurs de réputation doivent être robuste aux attaques ;
4. Un unique témoignage doit avoir peu d'influence sur le score de réputation d'un fournisseur.

À ces quatre propriétés, Carrara et Hogben rajoutent le fait qu'un moteur de réputation ne doit pas être trop complexe, pour permettre à ses utilisateurs de bien comprendre la signification des scores de réputation, et ainsi prendre des décisions éclairées [CH07].

Finalement, Jøsang et Ismail expliquent qu'au delà de proposer de nouveaux moteurs de réputation, il est nécessaire de développer des mécanismes de propagation des scores de réputation robustes :

> Up to now, most of the researches have concentrated on developing of reputation engine [...] However, we would like to point out that development of robust and secure mechanisms for reputation propagation is a topic that deserves much more study. (Jøsang et Ismail [JI02])

Dans le cadre de cette thèse, nous ne nous intéressons pas directement aux moteurs de réputation. Nous proposons un mécanisme de réputation qui s'articule autour d'un moteur de réputation pris en paramètre et lui permet de disposer des informations nécessaires au calcul robuste de scores de réputation. Afin de déterminer quelles sont ces informations, il faut étudier quelles attaques peuvent toucher les mécanismes de réputation, et quelles solutions permettent de les éviter. Par exemple, il faut s'assurer que les témoignages transmis au moteur de réputation soient légitimes, c'est-à-dire qu'ils sont issus d'une transaction ; sinon, n'importe quel client peut se faire passer pour un témoin et modifier – en augmentant ou en diminuant – la réputation d'un fournisseur donné.

3.3 Attaques sur les mécanismes de réputation

De nombreux travaux se sont intéressés aux attaques possibles sur les mécanismes de réputation. En effet, nous avons expliqué précédemment qu'une bonne réputation donne un avantage économique (voir section 3.1) ; attaquer un mécanisme de réputation peut ainsi permettre d'obtenir un avantage économique tout en fournissant un service déplorable. Nous présentons ici ces attaques, en décrivant les méthodes permettant de les limiter, ou même de les empêcher.

3.3.1 Blanchiment de réputation

La première attaque mise en avant est l'attaque par *blanchiment* [CH07 ; Del06 ; HZN09 ; JIB07]. Cette attaque se produit lorsqu'un fournisseur de mauvaise réputation se crée un nouveau compte pour repartir de zéro. Ainsi, la réputation du fournisseur devient la réputation initiale. Si celle-ci est trop élevée, le fournisseur est capable de tromper ses nouveaux clients. Pour empêcher cette attaque, il est recommandé d'assigner une réputation initiale faible aux nouveaux arrivants. Cependant, si cette réputation initiale est trop faible, les nouveaux arrivants ne pourront jamais attirer de clients et ne vont donc pas s'intégrer au système. Il faut donc trouver un compromis entre la résistance au blanchiment et l'intégration des nouveaux arrivants. Une autre solution consiste à demander un coût lors de chaque inscription, qu'il soit monétaire ou calculatoire [Bor06] ; la capacité de blanchiment d'un fournisseur de service est alors limitée par sa capacité monétaire ou calculatoire.

3.3.2 Discrimination

Une deuxième attaque possible concerne les témoignages « injustes » et, plus généralement, les discriminations [CH07 ; Del00 ; HZN09 ; JIB07 ; Liu+11 ; WJI04]. En effet, même si Resnick

et Zeckhauser montrent que les mécanismes de réputation fonctionnent bien et que les clients sont majoritairement satisfaits [RZ02], rien n'empêche un client d'assigner systématiquement une mauvaise note à une certaine catégorie de fournisseurs, même lorsqu'ils se comportent bien – ou inversement. En fait, il y a six principales discriminations possibles, qui peuvent être regroupées en trois types :

1. un client notant tous les fournisseurs normalement, sauf un qu'il note positivement – du point de vue d'un observateur, ce cas est équivalent à un fournisseur qui se comporte correctement avec un seul client ;
2. un client notant tous les fournisseurs normalement, sauf un qu'il note négativement – du point de vue d'un observateur, ce cas est équivalent à un fournisseur qui se comporte incorrectement avec un seul client ;
3. un client notant tout fournisseur qu'il connaît positivement, ce qui est équivalent à un client notant négativement tout fournisseur qu'il ne connaît pas, et à un fournisseur se comportant correctement avec les clients qu'il connaît – ou incorrectement avec ceux qu'il ne connaît pas.

Notons que le cas où un fournisseur de service se comporte incorrectement avec des clients qu'il ne connaît pas n'est pas considéré ; dans ce cas, la réputation du fournisseur montrera qu'il n'est pas correct.

Une première solution consiste à garantir l'anonymat des utilisateurs : si un client ne peut pas distinguer deux fournisseurs, il est incapable d'en discriminer un par rapport à l'autre. Ainsi, cette solution résout les deux possibilités du deuxième cas, puisqu'un client *ne peut pas* cibler un fournisseur, et réciproquement. Cependant, cette solution n'empêche pas les deux autres types de discriminations : un client interagissant avec un fournisseur qu'il connaît peut se signaler afin que le fournisseur se comporte correctement, et réciproquement.

Pour empêcher ces attaques, il est possible de filtrer les témoignages jugés « injustes » [Liu+11 ; WJI04]. Deux approches différentes sont utilisées pour juger si un témoignage est « injuste ». La première, proposée par Whitby et coll. [WJI04], s'applique lorsque la réputation est modélisée par un système bayésien, c'est-à-dire par une loi de probabilité bêta (voir section 3.2.2). Cette méthode exploite les propriétés statistiques de la loi bêta, et plus précisément ses quantiles. Un quantile d'une distribution est une valeur en dessous de laquelle se trouve une fraction donnée de la distribution ; par exemple, le quantile de 5 % est la valeur q_5 telle que $P(X \leqslant q_5) = 5\,\%$. L'idée derrière le filtrage de Whitby et coll. est la suivante : si les quantiles de la loi bêta engendrée par les témoignages d'un client particulier sont trop éloignés de la réputation calculée à partir de tous les témoignages – représentés par l'ensemble T_{FS} –, alors ce client est filtré. Ce filtrage est réitéré jusqu'à ce que l'ensemble des témoignages filtrés obtenu F_{FS} soit stable. L'algorithme 3.1 détaille cette procédure. Liu et coll. [Liu+11] proposent d'utiliser l'algorithme de partitionnement de données DBSCAN [Est+96] pour regrouper les témoins en grappes en fonction de leurs témoignages. Tous les témoins déviant de la plus grosse grappe sont jugés « injustes », et sont donc écartés. Ainsi, les cas 1 et 3 sont détectés : si un client note systématiquement positivement un fournisseur particulier alors qu'il se comporte incorrectement avec la plupart de ses clients, les témoignage positifs seront jugés injustes. Similairement, les témoignages injustement négatifs sont détectés.

$F_{FS} \leftarrow T_{FS}$
repeat
  $\text{rep}_{FS} \leftarrow$ réputation calculée à partir de F_{FS}
  for all Client Cl de F_{FS} **do**
    $\text{rep}_{Cl} \leftarrow$ loi bêta calculée à partir des témoignages de Cl
    $q_5 \leftarrow$ quantile de 5 % de rep_{Cl}
    $q_{95} \leftarrow$ quantile 95 % de rep_{Cl}
    if $q_5 > \text{rep}_{FS}$ ou $q_{95} < \text{rep}_{FS}$ **then**
      $F_{FS} \leftarrow F_{FS} \setminus \{Cl\}$
until F_{FS} est stable

Listing 3.1 – Filtrage des témoignages injustes [WJI04]

Aucune de ces deux approches n'est idéale : l'anonymat des utilisateurs empêche uniquement le deuxième type de discriminations, tandis que le filtrage des témoignages jugés « injustes » permet de détecter les autres. Ces deux approches ont donc des fonctionnalités complémentaires. Est-il possible de les combiner pour empêcher toute discrimination ? Au premier abord, l'anonymat paraît incompatible avec les méthodes de filtrage, qu'elles soient statistiques ou par partitionnement. Dans les deux cas, ces méthodes nécessitent de regrouper les témoignages par client pour le filtrage. Bethencourt et coll. [BSS10] proposent une méthode garantissant l'anonymat des utilisateurs (propriétés 1 et 2), tout en permettant d'associer les témoignages sur un fournisseur (propriété 7). Cette méthode peut être utilisée pour combiner anonymat et filtrage des témoignages injustes.

3.3.3 Bourrage d'urne

Dellarocas présente les attaques dites par « bourrage d'urne » [Del00]. Pour mener une telle attaque, un client interagit de nombreuses fois avec un fournisseur ; à l'issue de chaque interaction, le client dépose un témoignage positif sur le fournisseur. Ainsi, si un client dépose une écrasante majorité de témoignages positifs, la réputation du fournisseur sera bonne quel que soit son comportement avec d'autres clients. Si les témoignages et les clients les ayant émis sont publics, il est facile d'empêcher de telles attaques [CH07 ; HZN09 ; JIB07] ; il est possible d'utiliser les techniques de filtrage décrites précédemment [Liu+11 ; WJI04], ou par exemple en ne prenant en compte qu'un seul témoignage en compte par client. Cependant, l'anonymat des utilisateurs exacerbe cette attaque. Heureusement, Bethencourt et coll. [BSS10] proposent une technique associant les témoignages sur un même fournisseur, ce qui permet d'en prendre un seul en compte par client, tout en préservant la vie privée des clients ; la section 3.5.2 présente ce mécanisme.

3.3.4 Attaque Sybil et collusions

L'attaque Sybil, décrite par Douceur [Dou02], est présente dans tous les systèmes ouverts. Cette attaque apparaît lorsqu'un utilisateur se crée de multiples comptes afin d'augmenter son influence dans un système. Dans le cas des mécanismes de réputation, cela permet d'amplifier

les effets des autres attaques [CH07 ; HZN09], par exemple des témoignages injustes ou des attaques par bourrage d'urne présentées précédemment. En effet, lors d'un bourrage d'urne classique, un seul client dépose des témoignages négatifs sur un fournisseur ; en combinaison avec une attaque Sybil, malgré les protections présentées précédemment, n témoignages sont pris en compte, où n est le nombre d'identités Sybil. Pour limiter cette attaque, les solutions sont similaires à celles visant à empêcher le blanchiment de réputation : exiger un coût, monétaire ou calculatoire [Bor06], lors de l'inscription d'un utilisateur empêche un utilisateur unique de se créer trop de comptes.

Hoffman et coll. insistent également sur le danger des collusions dans les systèmes distribués [HZN09]. Prenons le cas de EigenTrust [KSG03] ou du mécanisme proposé par Ravoaja et Anceaume [RA07] (voir section 3.2.4). Dans ces deux cas, la réputation d'un fournisseur de service est maintenue par un ensemble d'utilisateurs du système, appelés gestionnaires de score dans le cas de EigenTrust. Si une collusion est capable de contrôler une majorité des gestionnaires de score d'un fournisseur, alors elle contrôle la réputation de ce fournisseur, et est capable de la modifier arbitrairement. Pour empêcher une telle attaque, Hoffman et coll. recommandent de choisir les gestionnaires de score aléatoirement et en nombre suffisant ; de cette manière, la probabilité qu'une collusion contrôle suffisamment de gestionnaires de score pour modifier la réputation d'un fournisseur est faible. Notons que dans ces deux mécanismes, les gestionnaires sont choisis aléatoirement et déterminés par une DHT, ce qui rend ardu les collusions ; nous présentons en section 4.2 une analyse du nombre de gestionnaires nécessaire en fonction des paramètres du système et de la sécurité désirée.

3.3.5 Autres types d'attaques

Finalement, Carrara et Hogben présentent d'autres attaques pouvant toucher les mécanismes de réputation [CH07]. La première concerne les extorsions. Sur eBay, le client et le fournisseur se notent mutuellement ; le client peut donc menacer le fournisseur d'un témoignage négatif si le fournisseur ne dépose pas un témoignage positif. Pour empêcher de telles représailles, les auteurs recommandent de préserver la vie privée des utilisateurs, ou d'utiliser une mise en séquestre temporaire : si le client n'apprend la note du fournisseur qu'après avoir déposé la sienne, l'attaque est impossible.

La deuxième attaque concerne le vol de réputation et le déni de transaction. En effet, les mécanismes de réputation doivent garantir qu'un fournisseur de service ne peut voler la réputation d'un autre fournisseur – ce qui correspond à la propriété d'inforgeabilité des scores de réputation (propriété 6) –, et qu'un fournisseur ne peut empêcher un client de témoigner – ce qui correspond aux propriétés d'indéniabilité (propriétés 3 et 4).

Finalement, les auteurs rappellent les menaces relatives au réseau sous-jacent, et ne sont pas spécifiques aux mécanismes de réputation. Par exemple, le service peut être interrompu par une attaque de déni de service, empêchant un client de vérifier la réputation d'un fournisseur. Les informations relatives à la réputation des fournisseurs doivent également être sécurisés, pour empêcher qu'un adversaire ne les modifie.

3.3.6 Bilan

Comme nous avons pu le voir, de nombreuses attaques visent les mécanismes de réputation [CH07 ; Del00 ; HZN09 ; JIB07]. Tout d'abord, les fournisseurs peuvent blanchir leur réputation, pour revenir à une réputation initiale après avoir en avoir obtenu une mauvaise. Les clients et fournisseurs peuvent également se discriminer, en fournissant des services de mauvaise de qualité ou en émettant des témoignages injustes. Les utilisateurs peuvent aussi essayer de bourrer les urnes pour modifier les réputations, et mettre en œuvre des attaques Sybil pour amplifier les effets des attaques. De ces attaques, Jøsang et coll. jugent que les plus graves concernent le bourrage d'urne et les témoignages injustes :

> The problems of unfair ratings and ballot stuffing are probably the hardest to solve in any reputation system that is based on subjective ratings [...] (Jøsang et coll. [JIB07])

Cependant, des solutions existent pour éviter ces attaques, par exemple en imposant un coût à l'entrée du réseau [Bor06] ou en filtrant les témoignages jugés injustes [Liu+11 ; WJI04]. Il est intéressant de noter que préserver la vie privée des utilisateurs empêche plusieurs types d'attaques, comme les discriminations ou les extorsions. Et, bien que préserver la vie privée des utilisateurs gêne le filtrage des témoignages, des mécanismes innovants permettent de combiner ces deux solutions [BSS10].

Un mécanisme de réputation garantissant les propriétés de sécurité (propriétés 3 à 7) présentées au chapitre 1 est protégé des bourrages d'urnes, des vols de réputation et des dénis de réputation. De plus, les propriétés de vie privée (propriétés 1 et 2) empêchent un des types de discriminations. La protection contre les blanchiments et contre les attaques Sybil dépend de l'enregistrement des utilisateurs, tandis que la protection contre les autres types de discrimination dépend directement du moteur de réputation, et requiert la faculté d'associer les témoignages émis sur un même fournisseur, ce qui correspond à la propriété 7.

3.4 Réputation et vie privée

Pour accomplir leur objectif, les mécanismes de réputation requièrent des données pouvant donner des informations sur leurs utilisateurs, qu'ils soient clients ou fournisseurs. Mahler et Olsen expliquent que cette collecte d'informations peut inquiéter les utilisateurs :

> Introducing a reputation system requires a rather extensive collection, evaluation and disclosure of data. When deciding whether or not to participate in a reputation system, a potential user's concern may be whether the system will meet reasonable expectations with respect to privacy. Users may fear that too much information about them is collected and disseminated. (Mahler et Olsen [MO04])

À cet effet, Mahler et Olsen s'intéressent au cadre légal de la protection de la vie privée dans les mécanismes de réputation. Dans l'Union Européenne, ce cadre est régi par la directive sur la protection des données [Eur95]. Mahler et Olsen s'intéressent aux conséquences de cette directive sur la conception des mécanismes de réputation. Cette directive définit entre autres trois notions :

Donnée personnelle La directive définit une *donnée personnelle* comme « toute information liée à une personne naturelle identifiée ou identifiable », c'est-à-dire tout ce qui peut être perçu, senti ou enregistré à propos d'une personne naturelle – cela ne concerne donc pas les personnes morales.

Sujet de la donnée Le *sujet de la donnée* est la « personne naturelle » concernée par la donnée personnelle.

Contrôleur de la donnée Le *contrôleur de la donnée* est celui qui détermine les objectifs et moyens du traitement des données personnelles.

Bygrave explique que les opinions, même fausses, rentrent dans le cadre des données personnelles [Byg02, page 46]. Dans un mécanisme de réputation, les données personnelles sont donc les opinions des utilisateurs, c'est-à-dire les témoignages. Les sujets des données sont donc les témoins ayant émis les témoignages. Dans un mécanisme centralisé, le contrôleur des données est l'autorité centrale, tandis qu'il peut être constitué de plusieurs utilisateurs dans un mécanisme distribué. En plus de ces trois notions, Mahler et Olsen introduisent le *sujet de la réputation* pour désigner les fournisseurs de service ciblés par les témoignages [MO04]. Finalement, Mahler et Olsen décrivent neufs principes que doivent suivre les mécanismes de réputation afin de respecter la directive sur la protection des données. De ces neuf principes, les cinq principaux sont les suivant :

1. Les utilisateurs doivent donner leur consentement éclairé avant de témoigner ;
2. Le but des mécanismes de réputation doit être clairement défini ;
3. La collecte, le stockage et la dissémination des données doivent être limités aux éléments indispensables ;
4. La collecte des données et l'évaluation des scores de réputation doivent être transparentes et clairement expliquées aux utilisateurs ;
5. Les données personnelles doivent être sécurisées.

En respectant ces principes, Mahler et Olsen expliquent qu'un mécanisme de réputation améliore son utilité, et facilite son acceptation par les utilisateurs. Le troisième point est notamment intéressant, car il explique que les informations non nécessaires ne doivent pas être collectées ; c'est le cas de l'identifiant des clients.

Steinbrecher s'appuie sur cette analyse pour étudier plus en détails les propriétés de vie privée que les mécanismes de réputation doivent garantir à leurs utilisateurs [Ste08]. Steinbrecher rappelle néanmoins que les mécanismes de réputation ne doivent pas compromettre leur utilité afin de préserver la vie privée de leurs utilisateurs : un mécanisme de réputation préservant la vie privée ne sera pas utilisé s'il ne permet pas de calculer des scores de réputation résumant précisément les comportements des fournisseurs de service.

Selon Steinbrecher, les mécanismes de réputation doivent remplir plusieurs critères en terme de sécurité [Ste08] ; notamment, les témoignages doivent être inforgeables et les scores de réputation doivent être disponibles. Au niveau des propriétés de vie privée, Steinbrecher explique que les mécanismes de réputation doivent garantir deux propriétés principales, tirées des travaux de Pfitzmann et Hansen [PH10]. La première correspond à l'*anonymat* des utilisateurs, qui est défini par Pfitzmann et Hansen comme « [...] the state of being not identifiable within a

set of subjects, the *anonymity set.* » La seconde est l'*inassociabilité* des actions d'un utilisateur, définie par Pfitzmann et Hansen comme suit :

> *Unlinkability* of two or more items of interests [...] means that within the system [...], from the attacker's perspective, these items of interest are no more and no less related after his observation than they are related concerning his a-priori knowledge. (Pfitzmann et Hansen [PH10])

Par exemple, les fournisseurs de service doivent être anonymes dans l'ensemble des fournisseurs de même réputation ; le but des mécanismes de réputation est de distinguer les fournisseurs de service se comportant correctement de ceux qui se comportement incorrectement, l'ensemble d'anonymat des fournisseurs ne peut donc comprendre tous les fournisseurs. Il faut cependant faire attention à ce que les ensembles d'anonymats des fournisseurs soient suffisamment larges, par exemple en discrétisant les scores de réputation possibles des fournisseurs. Le moteur de réputation utilisé doit donc être adapté, et calculer les scores de réputation de manière à ce que l'ensemble d'anonymat des fournisseurs soit suffisamment large. D'un autre côté, les clients doivent être anonymes parmi tous les clients.

Dans un mécanisme de réputation, les objets d'intérêts sont les transactions et les témoignages des clients. La propriété d'inassociabilité signifie donc qu'un adversaire ne peut pas savoir si le même client participe aux mêmes transactions, ou a émis les mêmes témoignages. Nous avons expliqué précédemment (voir section 3.3.3) que, afin de se prémunir des bourrages d'urne, il est nécessaire de restreindre la propriété d'inassociabilité et d'être capable d'associer les témoignages ciblant un même fournisseur, à la manière du mécanisme de Bethencourt et coll. [BSS10].

Les propriétés de vie privée des fournisseurs et des clients (propriétés 1 et 2) que nous avons définies au chapitre 1 expriment ces deux critères, tandis que la propriété d'associabilité des témoignages (propriété 7) garantit que les bourrages d'urne peuvent être détectés, sans toutefois rentrer en conflit avec la vie privée des clients.

3.5 Mécanismes de réputation préservant la vie privée

Maintenant que nous avons expliqué pourquoi et de quelle manière les mécanismes de réputation doivent respecter la vie privée de leurs utilisateurs, nous nous intéressons aux mécanismes préservant la vie privée existants. Nous étudions ces mécanismes sous la lumière des propriétés de vie privée et de sécurité définies précédemment (voir section 1.4). Nous commençons par présenter les mécanismes de réputation préservant le secret des témoignages, qui furent les premiers à apparaître. Nous nous intéressons ensuite aux mécanismes offrant des garanties de vie privée plus évoluées. La table 3.2 présente les propriétés garanties par ces mécanismes.

3.5.1 Mécanismes de réputation préservant le secret des témoignages

Pavlov et coll. [PRT04], « Supporting Privacy in Decentralized Additive Reputation Systems » Les auteurs de cet article proposent un mécanisme de réputation distribué permettant aux clients d'émettre des témoignages positifs ou négatifs, tout en garantissant le secret des

notes. Les auteurs proposent d'utiliser la bêta réputation comme moteur de réputation [JI02] (voir section 3.2.2).

Ce mécanisme suppose qu'un client demandant la réputation d'un fournisseur peut être curieux, mais pas malveillant (voir section 1.3) ; ce mécanisme tolère les malveillances des autres utilisateurs.

Cet article s'intéresse à deux problèmes. Tout d'abord, comment choisir les témoins à partir desquels construire le score de réputation d'un fournisseur ? Et, une fois ces témoins choisis, comment calculer la réputation du fournisseur sans divulguer les notes des témoins ? Dans la suite, nous utilisons les notations suivantes :

— $N > 1$ représente le nombre de témoins possibles

— $0 < n < N$ désigne le nombre de témoins choisis pour calculer la réputation du fournisseur

— $b < N$ représente le nombre de témoins malveillants

Avant de détailler les schémas de sélection de témoins, les auteurs montrent que si un seul témoin est honnête, alors connaître le score de réputation du fournisseur de service et les notes des autres témoins permet d'inférer la note du témoin honnête. Ainsi, il faut garantir qu'au moins deux témoins honnêtes sont choisis. Le premier schéma de sélection de témoin proposé, reposant sur une marche aléatoire, offre cette garantie avec probabilité $(1 - \frac{1}{n}) \cdot \frac{N-b-1}{N-1}$. Cette garantie n'est toutefois pas très forte, et permet d'avoir une indication sur les témoignages des deux clients honnêtes. C'est pourquoi les auteurs proposent un second schéma de sélection de témoins, reposant également sur une marche aléatoire, garantissant avec forte probabilité, qu'au moins $n \cdot \frac{N-b-n}{N}$ témoins honnêtes sont choisis.

Une fois que les témoins sont choisis, il est possible d'utiliser leurs témoignages pour calculer le score de réputation du fournisseur de service. Pavlov et coll. proposent à cet effet trois algorithmes. Dans la suite, nous notons $T_1, \ldots, T_n$ les témoins choisis, dont les notes sont les $r_1, \ldots, r_n$.

Le premier algorithme proposé est celui de la somme sécurisée, également proposé par Clifton et coll. [Cli+02], dont le principe est le suivant. Le client choisit un grand nombre aléatoire r_0, qu'il envoie au premier témoin. Lorsque le témoin T_i reçoit la somme s_{i-1}, il ajoute son témoignage pour obtenir $s_i = s_{i-1} + r_i$. Le dernier témoin envoie la somme finale s_n au client, qui vaut donc $s_n = r_0 + \sum_i r_i$. Le client, ayant choisi r_0, est capable d'obtenir la réputation du fournisseur, c'est-à-dire la somme $\sum_i r_i$. Comme le nombre r_0 est grand par rapport aux notes r_i des témoins, celles-ci sont « noyées » et les sommes de témoignages ne donnent aucune information sur les témoignages. Cependant, cet algorithme a deux limitations. La première est qu'il ne tolère que les comportements curieux ; en effet, un adversaire malveillant peut ajouter – ou soustraire – n'importe quelle note pour améliorer ou diminuer la réputation du fournisseur. De plus, si deux témoins T_i et T_{i+2} partagent leurs informations, ils peuvent obtenir la note du témoin T_{i+1} : $s_{i+1} - s_i = r_{i+1}$.

Le deuxième algorithme proposé s'appuie sur le partage de secret présenté en section 2.3.1. Le secret du témoin T_i est r_i, et les parts sont $t_{i,1}, \ldots, t_{i,n}$ telles que $\sum_j t_{i,j} = r_i$. Le témoin T_i envoie ensuite $t_{i,j}$ au témoin T_j, pour tout j. Chaque témoin somme les notes reçues et envoie la somme au client ; le témoin T_i calcule $s_i = \sum_j t_{j,i}$. Le client peut ensuite obtenir la réputation du fournisseur en calculant $\sum_i s_i$. Dans cet algorithme, le partage de secret garantit

qu'aucun témoin ne connaît la note d'un autre témoin, sous la condition qu'il y a au moins deux témoins honnêtes. Cependant, un témoin malveillant est capable d'augmenter ou de diminuer arbitrairement le score de réputation du fournisseur en choisissant les $t_{i,j}$, et de modifier les témoignages des autres témoins en modifiant s_i.

Le dernier algorithme proposé par Pavlov et coll. permet de calculer la réputation du fournisseur en détectant les tentatives de manipulation des témoins malveillants. Cet algorithme utilise du partage de secret vérifiable ainsi que des preuves de connaissance à divulgation nulle de connaissance (voir sections 2.3 et 2.5). Une fois les témoins choisis, chacun d'entre eux génère des parts de sa note à partir du polynôme

$$p_i : x \mapsto r_i + \sum_{j=1}^{n} p_{i,j} x^j,$$

où les $p_{i,j}$ sont choisis aléatoirement. Le témoin T_i envoie la part $p_i(j)$ à T_j, ainsi qu'une preuve de la validité de cette part ; cette preuve garantit que la note du témoin appartient à l'ensemble des notes possibles, par exemple l'ensemble $\{0; 0{,}1; \ldots; 1\}$. Si une part est invalide, le témoin envoie la part et la preuve incriminées au client. Chaque témoin somme les parts valides reçues. En notant $q = \sum_i p_i$ le polynôme somme des polynômes p_i, cette somme est $q(i) = \sum_{j=1}^{n} p_j(i)$. Les témoins envoient finalement ces sommes au client. Le client a donc obtenu les parts du polynôme q. En effectuant une interpolation de Lagrange des sommes reçues, le client peut obtenir $q(0) = \sum_i p_i(0) = r_i$, c'est-à-dire la réputation du fournisseur. Grâce aux preuves de connaissance à divulgation nulle de connaissance, cet algorithme tolère les utilisateurs malveillants ; cependant, le traitement des parts invalides n'est pas clair, et l'algorithme ne spécifie pas ce qu'il se passe lorsqu'un témoin malveillant envoie une part valide à un témoin et une autre invalide à un autre témoin.

De manière générale, ces algorithmes ont plusieurs problèmes. Tout d'abord, au niveau de la sélection des témoins : un utilisateur n'ayant jamais interagi avec le fournisseur peut se faire passer pour un témoin, et être choisi en tant que tel. Ainsi, l'inforgeabilité des témoignages (propriété 5), telle que nous la définissons, n'est pas garantie. Cela implique que l'inforgeabilité des scores de réputation (propriété 6) n'est pas garantie non plus ; en effet, un faux témoin rajoute un faux témoignage, ce qui modifie le score de réputation du fournisseur. Ensuite, un fournisseur ne peut obtenir de preuve de transaction ; si un client ne désire pas témoigner, la transaction en question ne laisse aucune trace sur le score de réputation du fournisseur. Ainsi, la propriété d'indéniabilité des preuves de transaction n'est pas garantie (propriété 4). Finalement, ces algorithmes ne préservent que le secret des notes, et non pas la vie privée des clients et des fournisseurs (propriétés 1 et 2).

Hasan [Has10], « Privacy Preserving Reputation Systems for Decentralized Environments » Hasan propose quatre mécanismes de réputation distribués, offrant des garanties de vie privée et tolérant des comportements malveillants de plus en plus évolués. Les trois premiers mécanismes supposent que les fournisseurs de service maintiennent eux-mêmes leur liste de témoins sans l'altérer, tandis que le dernier s'affranchit de cette hypothèse. Nous utilisons les mêmes notations que précédemment : les témoins sont les $T_i, 1 \leqslant i \leqslant n$, et leurs notes sont les r_i.

Le premier mécanisme proposé correspond à la somme sécurisée, également proposée par Pavlov et coll. [PRT04] et par Clifton et coll. [Cli+02].

Le principe du second mécanisme, appelé « aller-retour », est le suivant. Le client commence par demander la liste des témoins au fournisseur. Le client initie le chemin aller en choisissant un nombre aléatoire r_0. Il choisit ensuite un témoin aléatoire, à qui il envoie la somme initiale $s_0 = r_0$ ainsi que la liste des témoins restants. Quand le témoin T_i reçoit la somme s_{i-1} et la liste des témoins restants, il ajoute sa note ainsi qu'un nombre aléatoire $t_i : s_i = s_{i-1} + r_i + t_i$. Il choisit ensuite comme témoin suivant le témoin en qui il a le plus confiance parmi les témoins restants. Une fois le chemin aller fini la somme obtenue est $s_n = r_0 + \sum_i r_i + \sum_i t_i$, et le retour commence à partir du dernier témoin. Chaque témoin calcule successivement $s_{n+i} = s_{n+i-1} - t_i$. Quand la somme revient au client, il lui suffit d'ôter r_0 pour obtenir $\sum_i r_i$, c'est-à-dire le score de réputation du fournisseur. Pour obtenir la note d'un témoin cible, il faut donc contrôler quatre témoins : les deux témoins entourant la cible lors du chemin aller, ainsi que les deux l'entourant lors du retour. Lors de l'aller et du retour, un témoin choisit ses successeurs ; en utilisant la confiance qu'a un témoin en ses prédécesseurs et successeurs, il est donc possible de quantifier la probabilité d'une brèche de vie privée. Cependant, cet algorithme suppose que les témoins ne sont pas malveillants.

À l'instar du deuxième mécanisme de Pavlov et coll. [PRT04], le troisième mécanisme proposé par Hasan et coll. [HBB12] repose sur du partage de secret. Ce mécanisme, appelé k-parts, dépend d'un paramètre k, nombre entier. Comme dans les deux mécanismes précédents, le client commence par obtenir la liste des témoins auprès du fournisseur de service. Le témoin T_i choisit ensuite jusqu'à k témoins en qui il a confiance ; k_i est le nombre de témoins choisis par T_i. T_i partage son secret, c'est-à-dire sa note r_i, en k_i parts ; comme dans le mécanisme de Pavlov, la somme des parts d'un témoin donne sa note. Une fois les parts générées, T_i les envoie aux témoins qu'il a choisi. Chaque témoin somme ensuite les parts reçues, et envoie cette somme au client. En additionnant les sommes, le client obtient la réputation du fournisseur. Grâce au partage de secret, la note du témoin T_i ne peut être récupérée que si les k_i témoins choisis par T_i sont en collusion. L'avantage de cette approche par rapport au deuxième mécanisme de Pavlov et coll. [PRT04] est qu'elle permet aux témoins de quantifier la probabilité d'une brèche de vie privée en fonction de la confiance qu'ils ont en leurs pairs.

Contrairement aux mécanismes précédents, le dernier mécanisme proposé par Hasan et coll. [Has+13] tolère les utilisateurs malveillants disruptifs. Pour arriver à ses fins, ce mécanisme utilise du partage de secret vérifiable, du chiffrement asymétrique homomorphe randomisé, ainsi que des preuves de connaissance à divulgation nulle de connaissance. Ainsi, chiffrer un message requiert un aléa, et le chiffrement est homomorphe, c'est-à-dire qu'il possède la propriété suivante :

$$\forall k, x, y, \quad \mathrm{Enc}_k(x, \_) + \mathrm{Enc}_k(y, \_) = \mathrm{Enc}_k(x + y, \_)$$

Les auteurs proposent d'utiliser le cryptosystème de Paillier [Pai99], qui répond à ces exigences. Ce mécanisme requiert deux types de preuves de connaissance : le premier type permettant de prouver que le texte clair caché par un chiffré appartient à un ensemble, et le deuxième permettant de montrer que deux chiffrés sont issus du même texte clair. C'est-à-dire qu'en prenant X et Y des utilisateurs, un ensemble public E, un message $m \in E$, des aléas $r_1, r_2 \xleftarrow{R} \mathbb{Z}_p$,

et deux chiffrés $c_1 = \mathrm{Enc}_X(m, r_1)$ et $c_2 = \mathrm{Enc}_Y(m, r_2)$ de m, ces preuves s'écrivent sous la forme suivante :

$$\begin{aligned}
&\textsc{nizk}\left\{m : \mathrm{Dec}_X(c_1) = m \in E\right\},\\
&\textsc{nizk}\Big\{m, r_1, r_2 :\\
&\qquad c_1 = \mathrm{Enc}_X(m, r_1)\\
&\qquad c_2 = \mathrm{Enc}_Y(m, r_2)\Big\}.
\end{aligned}$$

Afin d'éviter qu'un fournisseur n'ajoute ou n'enlève un témoin, ce mécanisme assigne un ensemble de gestionnaires de score à chaque fournisseur de service, dont l'objectif est de maintenir l'ensemble des témoins ; cette approche est inspirée des mécanismes de réputation reposant sur une architecture pair à pair [KSG03 ; RA07] (voir section 3.2.4). Cette méthode garantit que, tant que la proportion de gestionnaires de score malveillants est faible, un adversaire ne peut pas modifier la liste des témoins, que ce soit en en ajoutant ou en en enlevant un.

Lorsqu'un client désire calculer la réputation d'un fournisseur, il commence par récupérer la liste des témoins des gestionnaires de score. Similairement au mécanisme précédent, chaque témoin choisit ensuite $k_i \leqslant k$ témoins de confiance parmi cette liste, notés $T_{i,1}, \ldots, T_{i,k_i}$. Le témoin T_i génère ensuite k_i parts $p_{i,1}, \ldots, p_{i,k_i}$ comme précédemment, qu'il chiffre de deux manière : avec sa propre clé publique, et avec la clé publique du témoin à qui il envoie la part :

$$\begin{aligned}
c^*_{i,j} &= \mathrm{Enc}_{T_i}(p_{i,j}, _),\\
c_{i,j} &= \mathrm{Enc}_{T_{i,j}}(p_{i,j}, _),
\end{aligned}$$

pour $1 \leqslant j \leqslant k_i$. Il calcule également la somme des parts chiffrées avec sa clé publique, c'est-à-dire $\beta_i = \sum_j c^*_{i,j}$. De par la propriété homomorphe du chiffrement, cette somme est également $\beta_i = \mathrm{Enc}_{T_i}(\sum_j p_{i,j}, \_)$, c'est-à-dire la note de T_i. Chaque témoin génère ensuite deux types de preuves : (a) une preuve que le texte clair caché par β_i, c'est-à-dire r_i, appartient à l'ensemble des notes possibles, et (b) k_i preuves que les chiffrés $c^*_{i,j}$ et $c_{i,j}$ sont issus du même texte clair. Ces preuves sont les suivantes :

$$\begin{aligned}
\pi_{\beta_i} &= \textsc{nizk}\left\{r_i : \mathrm{Dec}_{T_i}(\beta_i) = r_i \in E\right\},\\
\pi_{c_{i,j}} &= \textsc{nizk}\Big\{r_i :\\
&\qquad\qquad c^*_{i,j} = \mathrm{Enc}_{T_i}(r_i, _)\\
&\qquad\qquad c_{i,j} = \mathrm{Enc}_{T_j}(r_i, _)\Big\},
\end{aligned}$$

pour $1 \leqslant j \leqslant k_i$. Le témoin envoie les parts chiffrées ainsi que les $k_i + 1$ preuves au client. Le client peut ainsi vérifier la validité des parts et du témoignage, sans avoir plus d'information. Une fois les preuves vérifiées, le client envoie les parts chiffrées aux témoins correspondants. Le témoin T_i reçoit donc les $\{c_{i,j}\}_j$. Une fois ces chiffrés reçus, T_i calcule $\gamma_i = \sum_j c_{j_i}$, c'est-à-dire la somme des parts, chiffrée pour T_i. T_i déchiffre cette somme, et la rechiffre pour le client :

$$\gamma^*_i = \mathrm{Enc}_{\mathrm{Cl}}\left(\mathrm{Dec}_{T_i}(\gamma_i) = \textstyle\sum_j p_{j,i}, \_\right).$$

Finalement, T_i calcule une preuve que γ_i^* et γ_i sont issus du même texte clair :

$$\pi_{\gamma_i} = \text{NIZK}\Big\{ \textstyle\sum_j p_{j,i} : \gamma_i^* = \text{Enc}_{\text{Cl}}(\sum_j p_{j,i}, \_) \; \gamma_i = \text{Enc}_{T_i}(\sum_j p_{j,i}, \_)\Big\},$$

et envoie γ_i^* et π_{γ_i} au client. Avec cette preuve, le client s'assure que γ_i^* est bien issu de la somme des parts envoyées au témoin. Le client est finalement capable de déchiffrer chacune des sommes de parts, et d'obtenir la réputation du fournisseur.

Lors du déroulement de cet algorithme, les preuves de connaissance empêchent un témoin malveillant d'émettre un témoignage en dehors de l'intervalle prédéfini, ou de modifier les parts – et donc les témoignages – des autres témoins. Ainsi, ce mécanisme garantit l'inforgeabilité des témoignages et des scores de réputation (propriétés 5 et 6). De plus, les gestionnaires de score empêchent qu'un adversaire ne réduise un témoin au silence ; l'indéniabilité des notes est également garantie (propriété 3). Cependant, les fournisseurs ne peuvent obtenir de preuve de transaction de la part des témoins ne désirant pas témoigner.

Kerschbaum [Ker09], « A Verifiable, Centralized, Coercion-Free Reputation System » Kerschbaum propose un mécanisme de réputation bicéphale, permettant aux clients d'émettre des témoignages positifs et négatifs sans que personne ne connaisse les notes émises. Le but de ce mécanisme est d'empêcher les médisances, c'est-à-dire le deuxième type de discriminations présenté précédemment (voir section 3.3.2). Ce mécanisme utilise du chiffrement asymétrique homomorphe ainsi que des couplages pour atteindre cet objectif. La figure 3.2 décrit le fonctionnement de ce mécanisme ; dans cette figure, SP_1 et SP_2 sont les deux autorités centrales, Alice est le fournisseur de service et Bob est le client. Le fonctionnement est le suivant :

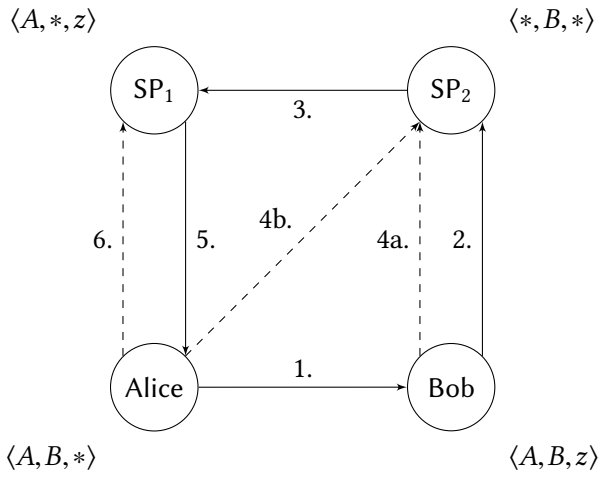


FIGURE 3.2 – Architecture du mécanisme de réputation [Ker09] [5]

5. Cette figure corrige une erreur de la figure présentée dans l'article : le $\langle A, *, z\rangle$ de SP_1 remplace le $\langle A, B, *\rangle$ puisque SP_1 connaît la note mais pas le client.

1. Alice donne un jeton à Bob avant d'effectuer la transaction ; ce jeton permettra à Bob d'émettre un témoignage. Le jeton permet à SP_1 d'identifier Alice, mais pas à SP_2. Une fois que Bob a reçu le jeton, Alice et Bob peuvent effectuer la transaction.
2. Bob émet son témoignage auprès de SP_2 ; ce témoignage comporte la note, qui est chiffrée de manière homomorphe pour SP_1, ainsi que le jeton.
3. SP_2 attend d'avoir reçu suffisamment de témoignages avant de les publier ; pour chaque témoignage, il publie le jeton ainsi que le chiffré de la note. Les éléments publiés permettent uniquement d'identifier Bob.
4. Le jeton permet à Bob de vérifier que son témoignage est publié, et à Alice de vérifier qu'aucun témoignage illégitime n'a été ajouté.
5. Pour chaque fournisseur de service, SP_1 rassemble les témoignages, déchiffre les notes et les utilise pour calculer le score de réputation. SP_1 calcule également une preuve de connaissance à divulgation nulle de connaissance du calcul des scores de réputation grâce aux propriétés homomorphes du schéma de chiffrement utilisé.
6. Finalement, Alice vérifie que SP_1 a bien calculé son score de réputation comme stipulé grâce à la preuve de connaissance.

Comme le montre la figure 3.2, Bob est le seul à connaître tous les éléments : Alice, Bob, ainsi que la note émise. De son côté, Alice ne connaît pas la note. SP_2 ne connaît que Bob, tandis que SP_1 ne connaît que Alice et la note. Ainsi, si SP_1 est en collusion avec Alice ou avec SP_2, ils sont capables de connaître Bob *et* la note émise.

Ce mécanisme, centralisé, permet aux clients d'émettre des témoignages positifs et négatifs. Il garantit que les témoignages émis sont légitimes, et que les scores de réputation sont correctement calculés. Cependant, il ne permet pas aux fournisseurs d'obtenir de preuve de transaction, et ne préserve la vie privée ni des clients, ni des fournisseurs. De plus, l'autorité calculant les réputations, SP_1, ne sait pas quels clients émettent les témoignages ; cette autorité ne peut donc pas associer les témoignages pour empêcher les bourrages d'urne. Un autre problème se pose pour les fournisseurs de service effectuant peu d'interactions. En effet, il est crucial que SP_2 publie *plusieurs* témoignages simultanément pour garantir le secret de la note, sans quoi les fournisseurs peuvent inférer la valeur des témoignages reçus à partir des variations du score de réputation : si la réputation a diminué après une mise à jour de quelques témoignages, le fournisseur peut en déduire que les clients ont émis des notes plutôt négatives. Ainsi, il faut beaucoup de temps pour mettre à jour les scores de réputation de ces fournisseurs.

Goodrich et Kerschbaum [GK11], « Privacy-Enhanced Reputation-Feedback Methods to Reduce Feedback Extortion in Online Auctions » Goodrich et Kerschbaum proposent d'enrichir le mécanisme proposé par Kerschbaum [Ker09] afin qu'il résiste aux extorsions (voir section 3.3.5). Tout d'abord, ils proposent d'étendre ce mécanisme pour le rendre symétrique : le client note le fournisseur, et le fournisseur note également le client. Comme expliqué précédemment, Goodrich et Kerschbaum arguent qu'utiliser tous les témoignages disponibles sur un fournisseur permet d'avoir une information sur ces témoignages. C'est pourquoi ils proposent d'*échantillonner* les témoignages sur un fournisseur pour calculer son score de réputation. C'est-à-dire que le score de réputation d'un fournisseur est calculé à partir de $r < n$

témoignages, où n est le nombre de témoignages sur ce fournisseur. Les r témoignages sont choisis via un tirage avec remplacement. Les auteurs montrent que $r = n/2$ suffit à avoir une estimation précise du score de réputation des fournisseurs, tout en limitant le risque de brèche de vie privée.

3.5.2 Mécanismes préservant la vie privée de leurs utilisateurs

Schiffner et coll. [SPT11], « On the Limits of Privacy in Reputation Mechanisms » Schiffner et coll. proposent une formalisation des mécanismes de réputation et des propriétés de vie privée. Ils modélisent les mécanismes de réputation de la manière suivante :

- un ensemble d'utilisateurs, U ;
- un ensemble de pseudonymes possibles, P ;
- un ensemble de notes, M ;
- un ensemble de réputations, V.

Un témoignage est donc un élément de $U \times U \times M$ représentant le client, le fournisseur, et la note du client. L'historique des transactions est représenté par une séquence de témoignages $\mathcal{H} = \big((u_1, u'_1, m_1), \ldots, (u_n, u'_n, m_n)\big)$. Un mécanisme de réputation propose les quatre algorithmes suivants :

- Init, qui initialise les outils cryptographiques ainsi que l'historique $\mathcal{H}$;
- Rate : $U \times U \times M \to \emptyset$, qui ajoute un témoignage à $\mathcal{H}$;
- NewPseudo : $U \to P$, qui assigne un pseudonyme à un utilisateur ;
- GetRep : $U \to P \times V$ qui assigne un pseudonyme et une valeur de réputation à un utilisateur, en fonction de l'historique.

Schiffner et coll. proposent l'expérience suivante pour modéliser les propriétés de vie privée. L'adversaire choisit deux séquences de transactions ; une de ces séquences, choisie aléatoirement, est jouée. La propriété de vie privée est garantie si et seulement si l'adversaire est incapable de déterminer, avec une probabilité non négligeable, laquelle de ces deux séquences a été jouée en connaissant uniquement les pseudonymes et leur réputation une fois la séquence jouée. Les propriétés des deux séquences déterminent la propriété exacte de vie privée qui est garantie ; si ces deux séquences sont quelconques, alors la propriété de vie privée la plus forte est garantie – Schiffner et coll. parlent de « Strong Anonymity ». Les auteurs définissent un treillis de propriétés de vie privée suivant les similarités entre les deux séquences : font-elles la même longueur ? Impliquent-elles les mêmes clients ou fournisseurs ?

Ils donnent également cinq exemples de mécanismes de réputation utilisant leur modélisation. De ces cinq exemples, un seul respecte la « Strong Anonymity ». Dans cet exemple, les notes sont uniquement positives ($M = \{m\}$), et les utilisateurs sont ordonnées par ordre décroissant de témoignages : l'utilisateur ayant reçu le plus de témoignages obtient le pseudonyme « 1 », et ainsi de suite – les ex æquo sont aléatoirement départagés.

Cette modélisation n'est cependant pas parfaite. Tout d'abord, cette modélisation considère qu'une interaction – de son début jusqu'à l'émission du témoignage – est atomique. En pratique, une interaction peut prendre quelques jours – c'est le cas lorsque la transaction consiste

en l'envoi d'un objet par la poste –, et un client peut mettre quelques jours pour témoigner. En outre, cette modélisation ne tient pas compte de la concurrence des interactions : à un instant donné, GetRep renvoie *un seul* pseudonyme par fournisseur. Ainsi, si un fournisseur propose deux services simultanément, les clients savent que les services sont proposés par le même fournisseur. Au contraire de la propriété de vie privée des fournisseurs (propriété 1), cette propriété ne garantit pas qu'un client n'a *aucune* information sur un fournisseur de service au moment où il émet son témoignage. De plus, la propriété de « Strong Anonymity » pour les clients empêche d'associer les témoignages (propriété 7) pour prendre en compte les bourrages d'urne, et permet donc à un fournisseur d'augmenter arbitrairement sa réputation. Ainsi, cette propriété est trop forte, et réduit l'utilité des mécanismes de réputation. Clauß et coll. [CSK13] soulèvent un autre problème : dans un mécanisme garantissant la « Strong Anonymity », si un utilisateur désire ne pas changer de pseudonyme, alors la « Strong Anonymity » n'est plus garantie. Notons en outre que le seul mécanisme proposé respectant la « Strong Anonymity » ne permet pas aux clients d'émettre des témoignages négatifs.

En plus de ces problèmes, cette modélisation des mécanismes de réputation ne précise pas le déroulement du calcul des scores de réputation, de la preuve de réputation ou de l'émission des témoignages. Comment un fournisseur convainc-il un client de sa réputation ? Comment les clients assignent-ils les témoignages aux fournisseurs appropriés ? Les auteurs ne précisent pas si ce mécanisme nécessite une autorité centrale, connaissant les liens entre les fournisseurs et les pseudonymes, dont le but est d'émettre les témoignages et de calculer les scores de réputation. Il est donc impossible d'analyser ce mécanisme en terme d'indéniabilité des témoignages, ou d'inforgeabilité des témoignages ou des scores de réputation (voir propriétés 3 à 6).

Clauß et coll. [CSK13], « k-Anonymous Reputation » Comme expliqué précédemment, Clauß et coll. montrent que la propriété de « Strong Anonymity » définie par Schiffner et coll. [SPT11] est faillible. Ils proposent donc de la remplacer par le k-anonymat des fournisseurs de service et le secret des notes. De manière informelle, le k-anonymat garantit que l'ensemble d'anonymat de tout fournisseur est au moins de taille k. Les auteurs se reposent sur le modèle des mécanismes de réputation proposé par Schiffner et coll. [SPT11] pour définir cette propriété et proposer un mécanisme de réputation la garantissant. Le k-anonymat est plus précisément défini par l'expérience suivante :

1. L'adversaire effectue une séquence de Rate [6]
2. L'adversaire effectue un GetRep[6], ce qui donne un ensemble de couples (V, P) de valeurs de réputations et de pseudonymes ;
3. L'adversaire effectue *un* Rate[6] ;
4. L'adversaire effectue un GetRep[6], ce qui donne un ensemble de couples (V', P') de valeurs de réputations et de pseudonymes ;

Le k-anonymat des fournisseurs est garanti si, pour deux couples (v, p) et (v', p') renvoyés par les deux GetRep, l'adversaire est incapable de savoir si p et p' appartiennent au même utilisateur avec une probabilité supérieure à $1/k$.

6. Voir le modèle proposé par Schiffner et coll. [SPT11] pour la définition de ces algorithmes, présenté p. 60.

Les auteurs proposent ensuite un mécanisme de réputation respectant cette propriété. Les paramètres de ce mécanisme sont le nombre d'utilisateurs n, le seuil d'anonymat $k \leqslant n$, le nombre de niveaux de réputation $\ell \ll n$, et un paramètre de sécurité pour le secret des notes $0 < o < 1$ – plus o est proche de 0, plus le secret des notes est préservé, mais moins les scores de réputation sont précis. Les fournisseurs sont divisés en trois catégories : ceux qui veulent être anonymes, dans l'ensemble U_{anon}, ceux qui veulent être identifiables, dans l'ensemble U_{id}, et ceux qui n'ont pas de préférence, dans l'ensemble U_{dc}. En plus des algorithmes définis par Schiffner et coll. [SPT11], les auteurs rajoutent un algorithme SetPseudonymPolicy, définissant les ensembles U_{anon}, U_{id} et U_{dc}. Dans ce mécanisme, l'algorithme NewPseudo, c'est-à-dire la génération des pseudonymes, commence par trier les utilisateurs par niveau de réputation, en créant trois ensembles pour chaque niveau de réputation i : $U_{\text{anon},i}$, $U_{\text{id},i}$ et $U_{\text{dc},i}$. Ensuite, pour chaque niveau de réputation i, les étapes suivantes sont exécutées :

1. Si $U_{\text{anon},i}$ est vide, alors aucun des utilisateurs de ce niveau n'aura de pseudonyme.
2. Si $|U_{\text{anon},i} \cup U_{\text{dc},i}| \leqslant k$, alors NewPseudo renvoie $\perp$.
3. Sinon, l'algorithme choisit aléatoirement $k - |U_{\text{anon},i}|$ utilisateurs de $U_{\text{dc},i}$, et génère des pseudonymes aléatoirement pour ces utilisateurs ainsi que pour les utilisateurs de $U_{\text{anon},i}$.

Ainsi, NewPseudo a soit choisi aléatoirement des pseudonymes pour au moins k utilisateurs de chaque niveau de réputation, soit renvoyé $\perp$ pour signifier son échec. Cet algorithme est ensuite utilisé pour calculer les scores de réputation des fournisseurs via l'algorithme GetRep :

1. Si aucun nouveau témoignage n'a été émis depuis le dernier GetRep, cet algorithme renvoie les mêmes valeurs que lors du dernier appel.
2. Sinon, pour chaque utilisateur u :
   a) l'algorithme construit l'ensemble M_u des témoignages émis sur u ;
   b) $M_{u,o}$ est construit en tirant $o \cdot |M_u|$ témoignages aléatoirement avec remplacement de M_u – c'est-à-dire en utilisant la méthode de Goodrich et Kerschbaum [GK11] (voir page 60), permettant de préserver le secret des témoignages ;
   c) la réputation de l'utilisateur u est $\text{rep}_u = \lceil \ell \cdot o \cdot \sum_{m \in M_{u,o}} m \rceil$.
3. L'algorithme utilise ensuite NewPseudo pour générer les pseudonymes ; si NewPseudo renvoie $\perp$, alors GetRep renvoie également $\perp$.
4. Sinon, GetRep renvoie les liens pseudonyme–réputation pour les utilisateurs anonymes, et utilisateur–réputation pour ceux qui ne le sont pas.

Cet algorithme renvoie les réputations des fournisseurs si et seulement si ceux qui désirent rester k-anonymes le peuvent.

Ce mécanisme de réputation préserve donc la vie privée des fournisseurs le désirant. De plus, les clients peuvent émettre des témoignages positifs ou négatifs et, comme ils ne sont pas anonymes, leurs témoignages peuvent facilement être associés (propriété 7) pour empêcher les bourrages d'urne. Cependant, ce mécanisme ne précise pas l'émission des témoignages : comment un client fait-il pour que son témoignage soit pris en compte, alors qu'il ne connaît le fournisseur que par un pseudonyme temporaire ? Ainsi, l'indéniabilité des témoignages (propriétés 3 et 4) ne peut pas être analysée. Le mécanisme ne précise pas non plus ce qu'est un

témoignage valide. Sans cette information, il est difficile de savoir si l'adversaire est capable de forger un témoignage (propriété 5) et, similairement, si un fournisseur malveillant est capable de forger un score de réputation (propriété 6). L'implémentation la plus simple d'un tel mécanisme reposerait sur une autorité centrale, calculant les scores de réputation des fournisseurs, générant leurs pseudonymes et récoltant les témoignages, ce qui entraîne un point unique de confiance et de défaillance. De plus, comme ce mécanisme repose sur le modèle de Schiffner et coll. [SPT11], il souffre des mêmes limitations : les interactions sont considérées atomiques, et les fournisseurs de service changent de pseudonymes uniquement lorsque l'algorithme GetRep est exécuté ; les interactions d'un fournisseur entre deux appels successifs de GetRep sont donc associables, ce qui limite leur vie privée : la propriété de vie privée des fournisseurs (propriété 1) que nous proposons exige qu'un client n'ait *aucune* information sur le fournisseur au moment où le client émet son témoignage, ce qui n'est pas le cas dans ce mécanisme.

Androulaki et coll. [And+08], « Reputation Systems for Anonymous Networks » Androulaki et coll. proposent le premier mécanisme de réputation préservant la vie privée des clients ainsi que des fournisseurs de service, suivant nos définitions. Ce mécanisme utilise plusieurs outils cryptographiques pour atteindre cet objectif : de l'argent électronique [CHL05], des accréditations anonymes [CL01], ainsi que des signatures aveugles [Oka92]. Pour arriver à ses fins le mécanisme de réputation proposé repose sur une banque centrale qui, même malveillante, est incapable de désanonymiser les utilisateurs.

Les auteurs considèrent tout d'abord que les communications entre les utilisateurs sont anonymes, ce qui peut par exemple être atteint en utilisant Tor [DMS04]. Ils supposent également que le nombre de témoignages qu'un utilisateur peut émettre est limité. Ainsi, chaque utilisateur reçoit à intervalles de temps réguliers un nombre de points à distribuer, que les auteurs appellent *repcoins*. Finalement, les auteurs supposent que n'importe quels utilisateurs peuvent être malveillants, et qu'ils peuvent même mettre en place des collusions, c'est-à-dire collaborer afin d'essayer de compromettre le mécanisme. La banque elle-même ne peut compromettre la vie privée des utilisateurs ; elle peut cependant empêcher le fonctionnement correct du mécanisme.

Brièvement, ce mécanisme fonctionne de la manière suivante. Les fournisseurs de réputation sont organisés dans des groupes de réputation. Par exemple, le groupe G_i contient les fournisseurs ayant reçu plus de 2^i repcoins – un fournisseur peut donc appartenir à plusieurs groupes de réputation. La banque maintient les scores de réputation des fournisseurs de service. Les fournisseurs utilisent des accréditations anonymes pour prouver leur réputation sans révéler leur identifiant. Pour pouvoir noter les fournisseurs de service, un client commence par retirer un portefeuille de repcoin auprès de la banque. Pour témoigner, un client donne une repcoin au fournisseur, connu uniquement sous un pseudonyme. Le fournisseur est ensuite capable d'ajouter cette repcoin à sa réputation de manière anonyme.

Les garanties de ce mécanisme sont multiples. Tout d'abord, si la banque est honnête, la conformité est garantie ; un utilisateur honnête est capable à la fois de retirer des pièces à hauteur de son compte, et de prouver sa réputation. Ensuite, ce mécanisme préserve la vie privée des clients et des fournisseurs, même si la banque est malveillante : un adversaire ne peut ni associer un pseudonyme à un utilisateur, ni savoir si deux pseudonymes sont issus

du même utilisateur. Le mécanisme garantit également qu'un ensemble d'utilisateurs ne peut pas dépenser plus de repcoins qu'il ne peut en retirer et que si une repcoin est dépensée deux fois, alors le coupable peut être retrouvé. Une collusion ne peut cependant pas faire accuser un utilisateur honnête de double dépense. Finalement, une collusion d'utilisateurs ne permet pas de montrer une réputation supérieure au maximum des réputation d'un des utilisateurs de la collusion.

Les accréditations anonymes permettent aux utilisateurs de recevoir des pseudonymes d'organisations, qui sont vérifiables publiquement. Un schéma d'argent électronique permet à des utilisateurs de retirer de l'argent d'une banque, et de payer d'autres utilisateurs ; un tel schéma permet de détecter les doubles dépenses. Notons que la banque ne peut rien apprendre sur les dépenses d'un utilisateur. Finalement, les signatures aveugles permettent à un signataire de signer un message sans connaître ni le message signé, ni la signature résultante.

La figure 3.3 présente le déroulement d'un témoignage ainsi que d'une preuve de réputation. Dans cette figure, M est un fournisseur de service dont le pseudonyme est P_M et U est un

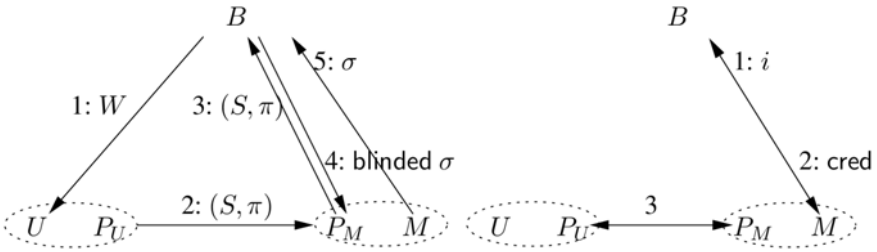


Figure 3.3 – Témoignage et preuve de réputation [And+08]

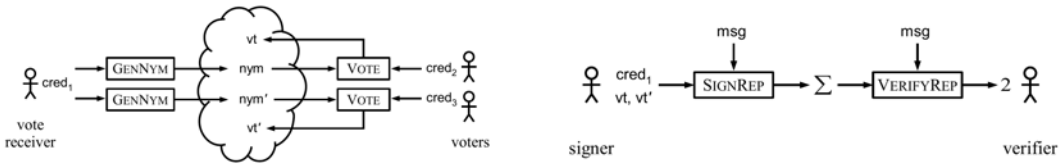
client, dont le pseudonyme est P_U. Avant que U ne puisse témoigner sur M, il faut que U retire un portefeuille W de la banque en utilisant le schéma d'argent électronique. U est ensuite capable de témoigner sur P_M en lui donnant une repcoin (S, π) ; S est le numéro de la repcoin, tandis que π en est une preuve de validité. Une fois que P_M a reçu la repcoin, P_M obtient une signature aveugle σ de la banque sur un message aléatoire, choisi par M, en déposant la repcoin. Finalement, M augmente sa réputation en déposant σ à la banque.

Une preuve de réputation se déroule en trois étapes. Tout d'abord, M s'enregistre auprès de la banque dans le groupe de réputation G_i. La banque vérifie que M a effectivement le droit d'intégrer le groupe G_i, et lui envoie une accréditation. Finalement, P_M prouve à P_U qu'il appartient au groupe G_i à l'aide d'une preuve de connaissance à divulgation nulle de connaissance.

Ce mécanisme préserve correctement la vie privée à la fois des clients et des fournisseurs de service : les clients apprennent uniquement la réputation des fournisseurs, tandis que les fournisseurs n'apprennent rien des clients. La banque elle-même est incapable d'associer un pseudonyme à une identité, ou deux pseudonymes. Ainsi, les propriétés de vie privée (propriétés 1 et 2) sont garanties. Cependant, la propriété de vie privée du client est trop forte ; en effet, la banque est incapable de savoir qu'un même client a émis deux repcoins, ce qui rend possible les bourrages d'urnes : l'associabilité des témoignages (propriété 7) sur un fournisseur n'est pas garantie. De plus, les clients ne sont pas capables d'émettre des témoignages négatifs. Finalement, le moteur de réputation utilisé n'est pas robuste : en effet, un fournisseur peut

transmettre sa réputation à un autre fournisseur en lui donnant les repcoins précédemment reçus. Ainsi, la réputation maximale d'une collusion de fournisseurs est la somme des réputations des fournisseurs et non pas la réputation du meilleur fournisseur de la coalition, ce qui est problématique.

Bethencourt et coll. [BSS10], « Signatures of Reputation » Bethencourt et coll. proposent un mécanisme de réputation distribué, où tous les utilisateurs sont anonymes. Ce mécanisme requiert une autorité centrale pour l'enregistrement des utilisateurs ; une fois les utilisateurs enregistrés, cette autorité n'est plus nécessaire. Comme le montre la figure 3.4a, les fournisseurs de service génèrent eux-même leurs pseudonymes. Après une interaction avec un fournisseur, les clients peuvent témoigner que le fournisseur s'est bien comporté, en émettant ce que les auteurs appellent un *vote*. Les fournisseurs peuvent ensuite agréger ces votes pour signer des messages avec leur réputation, que les clients peuvent vérifier avant d'interagir ; la figure 3.4b montre une telle preuve de réputation. Ce mécanisme permet également d'associer les témoignages sur un même fournisseur ; pour effectuer cette association, chaque témoignage comprend une valeur, l'*invariant*[7], dépendant uniquement de l'identité du fournisseur et de celle du client.



(a) Déroulement d'une interaction

(b) Preuve de réputation

FIGURE 3.4 – Signatures de réputation [BSS10]

Plus précisément, les propriétés garanties par ce mécanisme sont les suivantes. Tout d'abord, un client est incapable de savoir quel fournisseur émet une signature de réputation. Similairement, un client est incapable de savoir quel fournisseur reçoit son vote. Ensuite, si un client témoigne une seule fois sur le fournisseur, le fournisseur ne sait pas quel est ce client – si un client témoigne deux fois, alors le fournisseur est capable d'associer les deux témoignages, mais pas de savoir qui est ce client ou s'il a interagi avec un autre fournisseur. C'est-à-dire que ce mécanisme préserve les propriétés de vie privée définies au chapitre 1 (propriétés 1 et 2). Ce mécanisme est également conforme et sûr : un client honnête peut témoigner et un fournisseur honnête peut prouver sa réputation, tandis que les votes et signatures de réputation sont inforgeables (propriétés 5 et 6).

Pour atteindre ces objectifs, ce mécanisme emploie trois outils principaux : un système de preuves de connaissance non-interactives à divulgation nulle de connaissance [GS08] (voir section 2.5.2), un schéma de signatures courtes [BB08], ainsi que du chiffrement asymétrique homomorphe randomisé, dérivé du chiffrement linéaire proposé par Boneh et coll. [BBS04].

7. Bethencourt et coll. ne nomment pas cette valeur dans leur article, cependant elle est équivalente à l'invariant que nous introduisons dans le chapitre 5.

Un premier algorithme, Setup, génère les paramètres des outils cryptographiques utilisés. Le second, GenCred, permet à un utilisateur de générer ses clés et de s'enregistrer auprès de l'autorité centrale C pour recevoir un certificat. Chaque utilisateur possède quatre paires de clés : une clé de receveur rcvkey, servant à recevoir des votes, une paire de clés de votant $(\mathsf{votpk}, \mathsf{votsk})$, permettant de voter, une paire de clés de signature $(\mathsf{vk}, \mathsf{sk})$ et une paire de clés de chiffrement $(\mathsf{ek}, \mathsf{dk})$.

Les deux algorithmes suivants permettent de générer et de vérifier un pseudonyme. GenNym chiffre la clé de receveur avec la clé de chiffrement pour obtenir le pseudonyme nym, et construit une preuve de validité du pseudonyme en prouvant que l'utilisateur possède un certificat sur le texte clair du chiffré, ce qui se note : [8]

$$\begin{aligned}\textsc{nizk}\Big\{&\mathsf{cert}, \mathsf{rcvkey}, \mathsf{ek}, r : \\ &\mathsf{nym} = \mathrm{Enc}_{\mathsf{ek}}(\mathsf{rcvkey}, r) \\ &\wedge \mathrm{Verif}(\mathsf{cert}, \mathsf{rcvkey}, \mathsf{vk}_C)\Big\}.\end{aligned}$$

VerifNym vérifie la preuve pour s'assurer de la validité d'un pseudonyme.

Les algorithmes suivants permettent d'émettre et de vérifier un vote. Vote calcule l'invariant grâce au pseudonyme, et génère une preuve que l'invariant est bien calculé à partir du pseudonyme et de la clé de votant du client, c'est-à-dire :

$$\begin{aligned}\textsc{nizk}\Big\{&\mathsf{votesk}, \mathsf{votepk}, \mathsf{cert} : \\ &\mathsf{inv} = \mathsf{nym}^{\mathsf{votesk}} \\ &\wedge \mathsf{votesk} \text{ est la clé secrète associée à } \mathsf{votepk} \\ &\wedge \mathrm{Verif}(\mathsf{cert}, \mathsf{votepk}, \mathsf{vk}_C)\Big\}.\end{aligned}$$

VerifVote vérifie cette preuve pour garantir que le vote est émis par un client légitime.

Finalement, SignRep permet à un fournisseur de signer un message avec sa réputation. Notons $(\mathsf{inv}_i, \pi_i, \mathsf{nym}_i, \pi_{\mathsf{nym}_i})$ le i-ième vote, sa preuve, le pseudonyme l'ayant reçu et la preuve de validité de ce pseudonyme. L'algorithme commence par filtrer les votes pour n'en conserver qu'un par invariant différent, c'est-à-dire par client différent. Les invariants sont ensuite randomisés ; tous les invariants sont randomisés de la même manière pour conserver la propriété d'associabilité des votes : $\mathsf{inv}'_i = {\mathsf{inv}_i}^r$, où r est choisi aléatoirement. Le fournisseur génère également une paire de clés de signature, qu'il signe avec sa clé de signature : $\sigma_{\mathsf{ots}} = \mathrm{Sign}(\mathsf{sk}, \mathsf{vk}_{\mathsf{ots}})$. Le fournisseur construit ensuite une preuve garantissant que les votes sont valides, que les invariants sont correctement randomisés, que les pseudonymes sont valides, que tous les pseudonymes appartiennent au même fournisseur, et que la clé de signature est choisie par ce four-

8. Ceci est une version simplifiée des algorithmes de Bethencourt et coll. [BSS10], donnant l'intuition du fonctionnant de ce schéma.

nisseur :

$$
\begin{aligned}
\text{NIZK}\Big\{ & \mathsf{inv}_i, \pi_i, \mathsf{nym}_i, \pi_i', \mathsf{ek}, \mathsf{rcvkey}, r_i, \mathsf{cert}, r, \mathsf{vk}, \sigma_{\mathsf{ots}} : \\
& \forall i,\ \mathrm{Verif}(\pi_i, \mathsf{inv}_i) \\
\wedge & \forall i,\ \mathrm{Verif}(\pi_i', \mathsf{nym}_i) \\
\wedge & \forall i, \mathsf{nym}_i = \mathrm{Enc}_{\mathsf{ek}}(\mathsf{rcvkey}, r_i) \\
\wedge & \forall i, \mathsf{inv}_i' = {\mathsf{inv}_i}^r \\
\wedge & \quad \mathrm{Verif}(\sigma_{\mathsf{ots}}, \mathsf{vk}_{\mathsf{ots}}, \mathsf{vk}) \\
\wedge & \quad \mathrm{Verif}(\mathsf{cert}, \mathsf{rcvkey}, \mathsf{vk}_C)\Big\}.
\end{aligned}
$$

Notons que les pseudonymes et votes, bien que ne révélant pas l'identité du fournisseur, doivent être masqués : en effet, deux fournisseurs ayant eu les mêmes pseudonymes dans le passé sont en réalité le même fournisseur, ce qui impliquerait une brèche de vie privée. Finalement, le fournisseur signe le message du client ainsi que la preuve avec la clé de signature $\mathsf{sk}_{\mathsf{ots}}$ fraîchement générée. Une telle signature de réputation peut être vérifiée avec l'algorithme VerifRep.

Bethencourt présente également une analyse de la taille des différents éléments de ce mécanisme [Bet11]. Un pseudonyme fait environ 25 Kio, un vote 50 o, tandis qu'une signature de réputation sur n votes nécessite $n \cdot 500$ Kio. Il est possible de réduire la taille d'une signature de réputation en relâchant la propriété de sûreté. Les auteurs parlent alors de ε-sûreté, $0 \leqslant \varepsilon < 1$, qui garantit que si un fournisseur montre une signature de réputation sur n votes, alors le fournisseur a reçu au moins $(1 - \varepsilon) \cdot n$ votes. Les auteurs proposent une variante de signatures de réputation garantissant la ε-sûreté, en utilisant des arbres de hachages de Merkle [Mer89]. Dans ce cas, comme le montre la figure 3.5, la taille des signatures de réputation est réduite : pour $\varepsilon = 1/5$, la taille d'une signature converge vers 100 Mio.

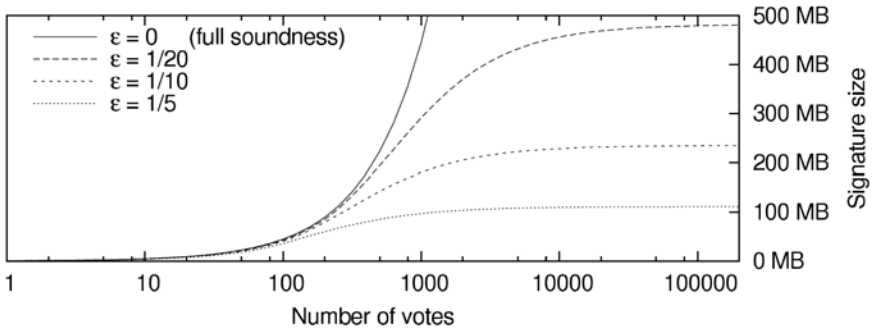


FIGURE 3.5 – Taille des signatures de réputation en fonction de ε [Bet11]

Nous avons expliqué que, grâce aux preuves de connaissances à divulgation nulle de connaissance, ce mécanisme de réputation préserve la vie privée des fournisseurs et des clients au sens des propriétés définies au chapitre 1 (propriétés 1 et 2). Ce système de preuve permet aussi de garantir l'inforgeabilité des témoignages et des scores de réputation (propriétés 5 et 6). Finalement, le calcul de l'invariant permet de garantir l'associabilité des témoignages (propriété 7).

Ce mécanisme a deux inconvénient principaux. Le premier est d'ordre pratique : les signatures de réputation requièrent un espace de stockage considérable. Même en considérant la ε-sûreté pour $\varepsilon = 1/5$, une signature de réputation nécessite 100 Mio ! Construire ou vérifier une telle signature demandera donc des puissances de calcul et de stockage importantes aussi bien aux clients qu'au fournisseur, ce qui n'est pas réaliste. Le deuxième inconvénient est plus problématique. En effet, les clients ne peuvent émettre que des témoignages positifs, ainsi la réputation des fournisseurs de service ne peut diminuer. Comme expliqué en section 3.1, un mécanisme de réputation doit permettre aux clients d'émettre des témoignages positifs et négatifs pour être utile.

3.5.3 Bilan

Dans cette section, nous avons commencé par présenter des mécanismes de réputation reposant uniquement sur des algorithmes distribués [Cli+02 ; Has10 ; PRT04]. Cependant, ces mécanismes ne garantissent que le secret des notes, supposent qu'aucun utilisateur n'est malveillant et ne tolèrent que des collusions de faible taille. Ces mécanismes ont alors intégré des outils cryptographiques, comme du chiffrement homomorphe ou du partage de secret, afin d'améliorer les garanties et tolérer les comportements malveillants [Has+13]. D'autres mécanismes, centralisés, garantissent également le secret des témoignages en dépit des comportements malveillants grâce à des outils cryptographiques [GK11 ; Ker09]. Des approches ont alors proposé une modélisation des mécanismes de réputation, et une définition formelle des propriétés de vie privée attendues [SPT11]. Ces propriétés sont toutefois imparfaites, et ne garantissent pas l'anonymat total des utilisateurs [CSK13], tandis qu'elles contredisent l'associabilité des témoignages et ne permettent pas d'empêcher les bourrages d'urne. Nous avons finalement décrit deux mécanismes préservant la vie privée de leurs utilisateurs. Le premier repose sur une autorité centrale, incapable de désanonymiser les utilisateurs [And+08] ; ce mécanisme ne permet toutefois pas d'associer les témoignages, et autorise les fournisseurs à transférer leur réputation à un autre utilisateur. Finalement, les signatures de réputation préservent la vie privée de leurs utilisateurs tout en associant les témoignages [BSS10]. Cependant, la collecte des témoignages utilisée par ces deux mécanismes ne garantit pas l'indéniabilité des témoignages, et ne permet donc pas aux clients d'émettre des témoignages négatifs. La table 3.2 décrit les caractéristiques de chacun de ces mécanismes de réputation.

3.6 Combiner réputation et respect de la vie privée

Dans ce chapitre, nous avons expliqué que les mécanismes de réputation ne préservant pas la vie privée des fournisseurs de service peuvent facilement stocker les témoignages et calculer leurs scores de réputation, par exemple en associant des gestionnaires de score à chaque fournisseur [Has+13 ; KSG03 ; RA07]. Les mécanismes préservant la vie privée des fournisseurs procèdent de deux manières : soit les fournisseurs transmettent les témoignages reçus à une autorité centrale, qui atteste ensuite de leur réputation [And+08], soit ils calculent eux-mêmes leur score de réputation et prouvent que le calcul est effectué à partir de témoignages valides [BSS10]. Dans les deux cas, l'indéniabilité des témoignages n'est pas garantie, ce qui explique que ces mécanismes ne tolèrent que des témoignages positifs. Il faut donc trouver un

Mécanisme	Distribué	Tém. +/-	Vie Privée		Indéniabilité		Inforgeabilité		Ass. tém
			FS	Client	notes	preuves	tém.	rép.	
Somme sécurisée [PRT04]	✔	✔	✗	✗	✗	✗	✗	✗	✔
Séparation du secret [PRT04]	✔	✔	✗	✗	✗	✗	✗	✗	✔
Partage de secret [PRT04]	✔	✔	✗	✗	✗	✗	✗	✗	✔
Aller-retour [Has10]	✔	✔	✗	✗	✗	✗	✗	✗	✔
k-parts [HBB12]	✔	✔	✗	✗	✗	✗	✗	✗	✔
k-parts malveillant [Has+13]	✔	✔	✗	✗	✔	✗	✔	✔	✔
Mécanisme bicéphale [Ker09]	✗	✔	✗	✗	✔	✗	✔	✔	✗
Réduction des extorsions [GK11]	✗	✔	✗	✗	✔	✗	✔	✔	✗
Exemple 5 [SPT11]	N.C.	✗	✗	N.C.	N.C.	N.C.	N.C.	N.C.	✗
k-anonymat [CSK13]	N.C.	✔	✗	✗	N.C.	N.C.	N.C.	N.C.	✔
Repcoin [And+08]	✗	✗	✔	✔	✗	✗	✔	✔	✗
Signatures de réputation [BSS10]	✔	✗	✔	✔	✗	✗	✔	✔	✔

TABLE 3.2 – Mécanismes de réputation préservant la vie privée [9]

moyen de combiner vie privée des fournisseurs et indéniabilité des témoignages (propriétés 1 et 3). Jusqu'à présent ce problème restait ouvert, comme le signalent Bethencourt et coll. :

> Most importantly, how can we support non-monotonic reputation systems, which can express and enforce bad reputations as well as good? Answering this question will require innovative definitions as well as cryptographic constructions. (Bethencourt et coll. [BSS10])

Nous avons également expliqué que la vie privée des clients doit être restreinte ; en effet, leur anonymat total permet d'effectuer des attaques par bourrage d'urne, comme dans le mécanisme proposé par Androulaki et coll. [And+08] ; à cet effet, il est possible d'utiliser la technique proposée par Bethencourt et coll. [BSS10], permettant d'associer les témoignages émis par un client sur un fournisseur donné.

Finalement, nous avons vu que certains mécanismes de réputation sont trop complexes pour permettre une mise en œuvre pratique, comme les signatures de réputation proposées par Bethencourt et coll. [BSS10]. Il faut donc veiller aux performances des mécanismes proposés.

Dans la suite, nous présentons comment construire un mécanisme de réputation distribué qui garantisse ces propriétés, tout en donnant la possibilité aux clients d'émettre des notes positives et négatives, et en permettant une implémentation efficace.

9. N.C. : non communiqué, la spécification du mécanisme de réputation ne précise pas ces aspects.

4 Préserver la vie privée des clients

Nous commençons par étudier uniquement la vie privée des clients (propriété 2), et proposons dans ce chapitre un mécanisme de réputation la garantissant. La description de ce mécanisme a été publiée à APVP 2012 ainsi qu'à ICC 2013 [Anc+13a ; Laj12].

Dans un premier temps, nous présentons une architecture de gestion des scores de réputation inspirée des architectures pair à pair existantes (voir section 3.2.4) qui permet de garantir l'intégrité des témoignages (propriétés 3, 4 et 5) – et donc des scores de réputation (propriété 6).

Nous détaillons ensuite le protocole d'interaction suivi par les clients et les fournisseurs de service, qui garantit l'indéniabilité des témoignages et des preuves de transaction (propriétés 3 et 4) tout en préservant la vie privée des clients.

Nous proposons également un moteur de réputation reposant sur la bêta réputation de Jøsang et Ismail [JI02], et adaptant le filtrage des témoignages de Whitby et coll. [WJI04] pour le rendre compatible avec l'anonymat des clients.

Finalement, nous discutons l'efficacité du moteur de réputation de deux manières : sur la précision théorique du score de réputation, et sur l'efficacité de l'adaptation de l'algorithme de filtrage à travers des simulations. Nous discutons également de la méthode utilisée pour préserver la vie privée des clients et proposons des pistes d'amélioration.

4.1 Objectifs

Dans ce premier mécanisme de réputation, nous nous fixons deux objectifs principaux. Tout d'abord, nous désirons préserver l'indistinguabilité entre les pseudonymes de deux clients. De plus, les témoins peuvent mentir sur le comportement réel des fournisseurs de service, ce qui n'est pas un comportement malveillant au sens de la définition 11. Nous nous intéressons donc à la précision du calcul des scores de réputation, malgré ces témoins menteurs.

À cet effet, nous considérons un système ouvert constitué de trois types d'utilisateurs : des clients, des fournisseurs et des gestionnaires de score qui, à l'instar de ceux utilisés par EigenTrust [KSG03] ou Ravoaja et Anceaume [RA07], maintiennent les réputations des fournisseurs (voir section 3.2.4). Un même utilisateur peut simultanément jouer les trois rôles, c'est-à-dire client, fournisseur et gestionnaire de score. Nous considérons également une autorité centrale C qui contrôle uniquement l'enregistrement des utilisateurs ; elle n'est plus nécessaire ensuite.

Nous avons expliqué précédemment que la réputation est contextuelle (voir section 3.1), c'est-à-dire qu'un fournisseur de service peut être très réputé dans un contexte particulier et beaucoup moins dans un autre. C'est pourquoi nous considérons deux types de réputation dans ce mécanisme, répondant aux questions suivantes :

— le fournisseur a-t-il fourni le service attendu par le client ?

— le fournisseur s'est-il comporté conformément au protocole d'interaction ?

Le second cas correspond à la notion d'honnêteté définie précédemment, tandis que le premier dépend intrinsèquement de la nature des services, et peut être plus subjectif.

4.2 Gestion des scores de réputation

La gestion des scores de réputation est un composant essentiel des mécanismes de réputation, qui garantit l'intégrité des témoignages. Nous présentons dans cette section une méthode de gestion des scores de réputation reposant sur une DHT, à l'instar de EigenTrust [KSG03] (voir section 3.2.4).

En général, les DHT assignent les identifiants des utilisateurs grâce à une fonction aléatoire à sens unique, qui renvoie une chaîne de m bits, par exemple à partir de l'adresse IP et du port utilisés par un utilisateur. En pratique, des fonctions de hachage cryptographiques sont utilisées pour garantir à la fois qu'un adversaire ne peut choisir son identifiant et que les collisions sont peu probables. Par exemple, pour la fonction SHA-256 [Nat12], l'identifiant est une chaîne de 256 bits, ce qui donne 2^{256} identifiants possibles. Une DHT définit une distance à partir des identifiants des utilisateurs ; par exemple, deux nœuds sont voisins si la longueur de leur préfixe commun est supérieure à un seuil donné. La figure 4.1 présente un exemple de DHT en anneau où $m = 4$, et où les voisins d'un nœud sont les trois nœuds le précédant sur l'anneau : les voisins du nœud 1 sont les nœuds 10, 12 et 15.

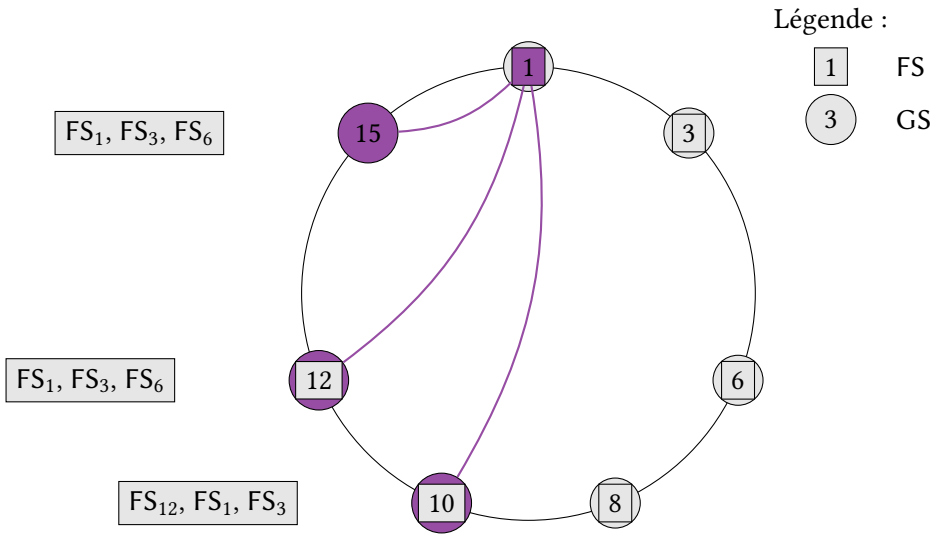


FIGURE 4.1 – Gestion des scores de réputation à travers une DHT

Dans notre mécanisme de réputation, les fournisseurs de service sont organisés dans une telle DHT. De cette façon, les gestionnaires de score d'un fournisseur donné sont ses voisins sur la DHT – d'autres fournisseurs ; cela permet de les retrouver simplement afin d'obtenir la réputation d'un fournisseur de service. En reprenant la figure 4.1, la réputation du fournisseur FS_1 est gérée par GS_{10}, GS_{12} et GS_{15}. Plus précisément, les gestionnaires de score d'un fournisseur donné stockent les témoignages émis sur ce fournisseur. Les clients potentiels d'un fournisseur

peuvent ainsi obtenir les témoignages émis sur un fournisseur aisément et, à partir de ces témoignages, calculer la réputation du fournisseur. La DHT gère également les arrivées et départs de nœuds. Lorsqu'un nœud intègre le système, il est assigné en tant que gestionnaire de score de plusieurs fournisseurs ; il se synchronise avec les autres gestionnaires de score de chacun de ces fournisseurs et commence à stocker les témoignages sur ces fournisseurs. Le départ d'un gestionnaire de score déclenche également une telle synchronisation. Afin d'éviter les attaques par bourrage d'urne, les gestionnaires de score ne conservent que les derniers témoignages émis par chaque pseudonyme.

Utiliser plusieurs gestionnaires de score pour un même fournisseur permet en outre de tolérer les comportements malveillants. En effet, nous montrons en section 4.3 que l'intégrité des témoignages est garantie tant que deux tiers des gestionnaires se comportent honnêtement. Le choix des gestionnaires de score doit donc garantir que la probabilité qu'il existe une collusion supérieure à un tiers soit faible. Cette probabilité dépend de plusieurs paramètres : N, le nombre de gestionnaires de score potentiels parmi lesquels ils sont choisis ; m, le nombre de malveillants parmi ces N utilisateurs ; n, le nombre de gestionnaires de score associés à chaque fournisseur ; et $f(n)$, la taille maximale d'une collusion parmi les gestionnaires de score choisis, qui dépend de leur nombre. Dans ce chapitre, $f(n)$ vaut $\lceil n/3 \rceil - 1$, mais nous utiliserons le même raisonnement ultérieurement pour $f(n) = \lceil n/2 \rceil - 1$.[1] En supposant que le choix des gestionnaires de score est aléatoire – ce qui est garanti par la DHT –, la probabilité d'obtention d'une collusion réussie est modélisée par la distribution hypergéométrique de paramètres N, m et n : il s'agit d'un choix sans remplacement de n éléments parmi N, dont m nous intéressent. Plus précisément, en notant $\mathrm{cdf}_{N,m,n}$ la fonction de répartition[2] de la distribution hypergéométrique de paramètres N, m et n, la probabilité de réussite d'une collusion est $p = 1 - \mathrm{cdf}_{N,m,n}\left(f(n)\right)$. Ainsi, si la probabilité maximale de réussite d'une collusion souhaitée est notée $p_{\max}$, le nombre de gestionnaires de score nécessaire n_{GS} est

$$n_{\mathrm{GS}} = \min\left\{n \in \mathbb{N} \mid 1 - \mathrm{cdf}_{N,m,n}\left(f(n)\right) < p_{\max}\right\}.$$

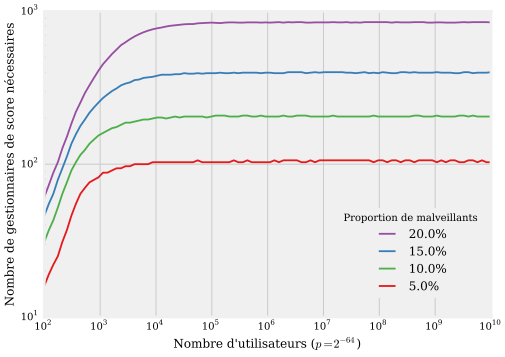
La figure 4.2 étudie n_{GS} pour $f : n \mapsto \lceil n/3 \rceil - 1$.

La figure 4.2a présente n_{GS} pour N compris entre 100 et 10^8, pour des proportions d'utilisateurs malveillants m/N fixées et pour $p_{\max} = 2^{-64}$, c'est-à-dire une probabilité extrêmement faible, qui va de pair avec la sécurité des outils cryptographiques utilisés. Cette figure montre que le nombre de gestionnaires de score nécessaire passe parfaitement à l'échelle : quelle que soit la proportion d'utilisateurs malveillants, le nombre de gestionnaires de score ne croît plus au delà de $N = 10^6$.

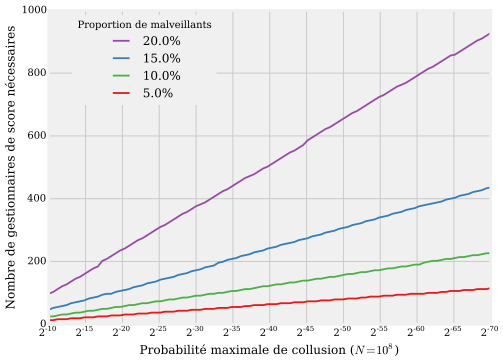
La figure 4.2b présente n_{GS} pour $N = 10^8$, pour $p_{\max}$ compris entre 2^{-10} et 2^{-70} et pour des proportions m/N fixées. Cette figure montre que le nombre de gestionnaires de score nécessaires est logarithmique en la probabilité de collusion désirée. Par exemple, $n_{\mathrm{GS}} = 100$ permet d'empêcher les collusions soit pour $m/N = 15\,\%$ avec une probabilité $p_{\max} = 2^{-20}$, pour $m/N = 10\,\%$ avec une probabilité $p_{\max} = 2^{-35}$, ou pour $m/N = 5\,\%$ avec une probabilité $p_{\max} = 2^{-20}$.

Finalement, la figure 4.2c présente n_{GS} en fonction de m/N, pour $N = 10^8$ et $p_{\max} = 2^{-64}$. Cette figure montre que, pour conserver un nombre de gestionnaires de score raisonnable,

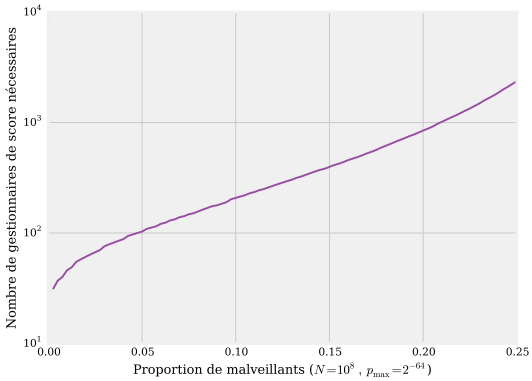
1. La section 4.3.3 explique que la synchronisation des gestionnaires de score pour l'acceptation d'un témoignage fonctionne seulement si le nombre de gestionnaires malveillants est inférieur à $\lceil n/3 \rceil - 1$.
2. *cumulative distribution function*



(a) En fonction du nombre d'utilisateurs



(b) En fonction de la probabilité maximale de collusion, pour $N = 10^8$



(c) En fonction de la proportion d'utilisateurs malveillants, pour $N = 10^8$ et $p_{\max} = 2^{-20}$

FIGURE 4.2 – Nombre de gestionnaires de score nécessaires pour rendre les collusions supérieurs à un tiers improbables

notre mécanisme peut utiliser 103 gestionnaires de score pour empêcher les collusions pour $m/N = 5\,\%$ avec une probabilité $p_{\max} = 2^{-64}$, ce qui rend extrêmement faible la probabilité que les gestionnaires de score d'un fournisseur du système soient en collusion.

Remarque

Deux fournisseurs de service proches sur la DHT peuvent se mettre en collusion pour essayer d'attaquer le mécanisme. Par exemple, reprenons l'exemple de la figure 4.1 et supposons que les fournisseurs FS_1 et FS_8 soient en collusion. FS_8 peut faire pression sur FS_{10} pour que ce dernier améliore la réputation de FS_1. Cependant, pour que la menace de FS_8 soit crédible, il faut que celui-ci contrôle suffisamment de gestionnaires de score de FS_{10} ; la probabilité d'un tel phénomène étant très faible, de telles attaques sont peu probables. ■

Dans la suite, nous supposons que moins d'un tiers des gestionnaires de score du fournisseur considéré sont malveillants.

4.3 Protocole d'interaction

Nous présentons maintenant les sous-protocoles du mécanisme d'interaction. Avant que les interactions puissent avoir lieu, il faut que le mécanisme ait été préparé, c'est-à-dire que les utilisateurs se soient enregistrés auprès de l'autorité centrale C pour obtenir leurs certificats et pseudonymes. Une interaction se divise ensuite en deux étapes : un client commence par obtenir la réputation du fournisseur qui l'intéresse. Une fois qu'il a trouvé un fournisseur lui convenant, il peut lancer l'interaction qui lui permet d'obtenir le service désiré et de témoigner.

4.3.1 Mise en place du protocole

Pendant une interaction, les utilisateurs requièrent des identifiants ainsi que des clés cryptographiques. Dans ce chapitre, nous utilisons le schéma de signature classique présenté en section 2.6.1. Tout d'abord, l'autorité centrale C possède une paire de clés de signature $(\mathsf{vk}_C, \mathsf{sk}_C)$, qui est utilisée pour émettre des certificats sur les utilisateurs.

Les fournisseurs génèrent également une paire de clés de signature : le fournisseur FS génère la paire $(\mathsf{vk}_{\mathsf{FS}}, \mathsf{sk}_{\mathsf{FS}})$, qu'il enregistre auprès de C pour obtenir un certificat $\mathsf{cert}_{\mathsf{FS}}$ sur $\mathsf{vk}_{\mathsf{FS}}$.

Afin de préserver leur vie privée, les clients fonctionnent différemment. Un client génère lui-même ses pseudonymes aléatoirement, ainsi qu'une paire de clés de signature associée à chaque pseudonyme. Pour le pseudonyme nym, le client génère la paire $(\mathsf{vk}_{\mathsf{nym}}, \mathsf{sk}_{\mathsf{nym}})$. Chaque pseudonyme est certifié par C, qui génère un certificat $\mathsf{cert}_{\mathsf{nym}}$ sur nym et $\mathsf{vk}_{\mathsf{nym}}$. Le nombre de pseudonymes de chaque client doit être limité pour éviter les attaques Sybil (voir section 3.3.4), qui amplifieraient sinon les bourrages d'urne possibles (voir section 3.3.3). Ainsi, C demande aux clients de payer un coût prédéfini lors de la certification d'un pseudonyme, que ce coût soit monétaire ou calculatoire [Bor06].

Dans la suite, nous notons nym le pseudonyme utilisé par le client, FS le fournisseur et GS_i ses gestionnaires de score, pour $1 \leqslant i \leqslant n$.

4.3.2 Obtention de la réputation d'un fournisseur

Quand un client est intéressé par le service proposé par un fournisseur, il contacte les gestionnaires de score de ce fournisseur à travers la DHT pour obtenir les témoignages émis sur ce fournisseur. Dès que le client a reçu les réponses de plus de deux tiers des gestionnaires, il effectue un quorum ; ceci empêche une collusion d'un tiers de gestionnaires malveillants de modifier les témoignages, et garantit donc leur inforgeabilité ainsi que celle des scores de réputation (propriétés 5 et 6). Le client est finalement capable de calculer la réputation du fournisseur comme spécifié par la section 4.4. La figure 4.3 décrit ce protocole.

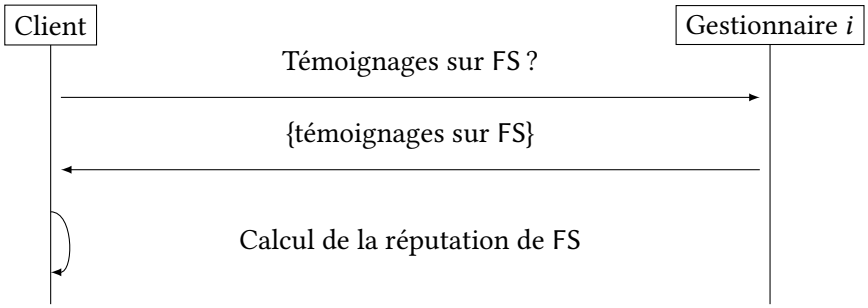


FIGURE 4.3 – Obtention des témoignages et calcul de la réputation du fournisseur

4.3.3 Interaction entre un client et un fournisseur

Une fois que le client a calculé la réputation du fournisseur, il choisit d'interagir avec le fournisseur s'il juge cette réputation suffisamment élevée. Cette interaction se déroule en quatre étapes : tout d'abord, le client et le fournisseur choisissent un identifiant de transaction. Ils signent cet identifiant et le transmettent, accompagné de la signature, aux gestionnaires de score du fournisseur. Une fois que les gestionnaires de score en ont confirmé la réception, la transaction peut avoir lieu. Finalement, le client et le fournisseur envoient leur note, qui permettent de constituer un témoignage rendant compte de la transaction. La figure 4.4 présente une telle interaction.

Le client commence par calculer l'identifiant de transaction $\mathrm{id}_{\mathrm{trans}}$ défini par

$$\mathrm{id}_{\mathrm{trans}} = H(\mathrm{nym}, \mathrm{FS}, \mathrm{timestamp}),$$

où H est une fonction de hachage classique comme SHA-256 [Nat12]. Cet identifiant permet de ne prendre en compte qu'un seul témoignage par transaction. Le client et le fournisseur signent cet identifiant en calculant

$$\begin{aligned} \sigma_{\mathrm{nym}} &= \mathrm{Sign}(\mathrm{sk}_{\mathrm{nym}}, \mathrm{id}_{\mathrm{trans}}) \\ \sigma_{\mathrm{FS}} &= \mathrm{Sign}(\mathrm{sk}_{\mathrm{FS}}, \mathrm{id}_{\mathrm{trans}}), \end{aligned}$$

qu'ils transmettent ensuite aux gestionnaires de score du fournisseur. De cette façon, les gestionnaires de score ont la preuve que le client et le fournisseur désirent effectuer la transaction. Une fois qu'un gestionnaire a reçu les deux signatures, il confirme leur réception au client et au fournisseur.

Après réception de plus de $(2n)/3$ confirmations des gestionnaires, le client et le fournisseur effectuent la transaction.

Une fois la transaction complétée, le client et le fournisseur émettent leurs notes ρ_{nym} et ρ_{FS} auprès des gestionnaires de score : les deux notes représentent l'opinion de leur émetteur sur le fournisseur. Comme la réputation s'intéresse aussi bien au service fourni qu'au comportement du fournisseur, les notes ont deux dimensions : la qualité du service fourni, ainsi que la conformité du comportement du fournisseur. Si, après un temps prédéfini, le client ou le fournisseur

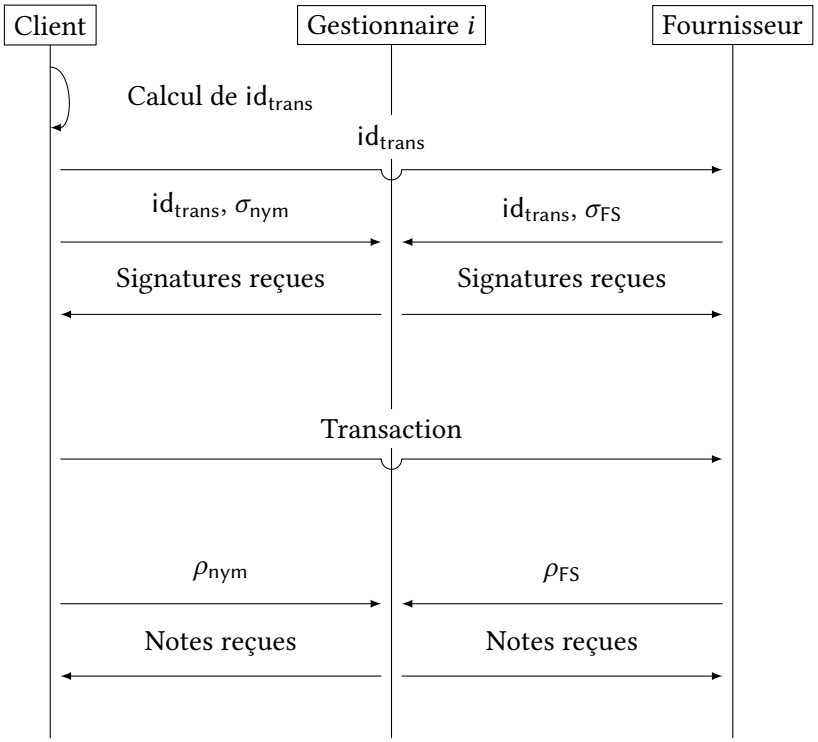


FIGURE 4.4 – Interaction entre un client et un fournisseur

n'ont pas émis de note, les gestionnaires assignent une note par défaut en considérant que les deux utilisateurs sont satisfaits : par défaut, leur note est 1.

Après réception – ou assignation par défaut – des deux notes, chaque gestionnaire de score utilise ces notes, le pseudonyme du client, ainsi que la date approximative pour créer un témoignage local. À partir de là, les gestionnaires utilisent un algorithme du consensus pour décider du témoignage conservé [LSP82], ce qui garantit que la synchronisation fonctionne tant que moins d'un tiers des gestionnaires sont malveillants ; ainsi, l'indéniabilité des témoignages (propriétés 3 et 4) est garantie. Plus précisément, Raynal [Ray02] explique qu'une solution au problème du consensus garantit trois propriétés :

Terminaison chaque gestionnaire de score honnête choisit une valeur ;

Validité faible Si tous les gestionnaires honnêtes proposent le même témoignage, alors chaque gestionnaire honnête choisit ce témoignage ;

Accord Les gestionnaires honnêtes choisissent tous un même témoignage.

Ainsi, si le client et le fournisseur envoient les mêmes notes à chaque gestionnaire de score, le témoignage accepté par les gestionnaires honnêtes est bien le témoignage émis par les utilisateurs. Cependant, si le client ou le fournisseur est malveillant et envoie des notes différentes aux gestionnaires de score, alors le témoignage résultant est indéterminé.

4.4 Moteur de réputation

Nous nous intéressons maintenant au moteur de réputation. La méthode proposée repose sur la bêta réputation de Jøsang et Ismail [JI02]. En plus de la valorisation et du vieillissement des témoignages, nous introduisons une étape de modulation, et nous adaptons la méthode de filtrage proposée par Whitby et coll. [WJI04] pour permettre de filtrer les témoignages émis de manière anonyme. Un témoignage est constitué de quatre informations : les notes du client et du fournisseur, ρ_{nym} et ρ_{FS} – des valeurs comprises dans l'intervalle $[0;1]$ –, la valeur de la transaction associée, v_ρ, ainsi que le moment auquel il a été émis, t_ρ. La figure 4.5 présente les quatre étapes du calcul du score de réputation.

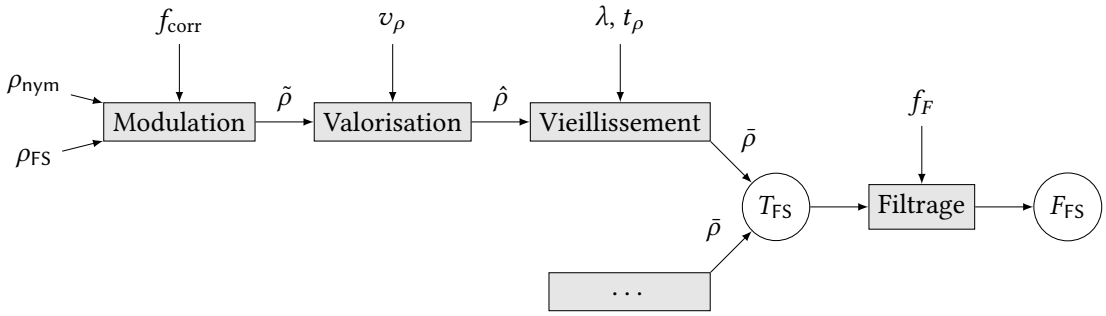


FIGURE 4.5 – Étapes du calcul des scores de réputation

Tout d'abord, les notes du client et du fournisseur sont modulées pour obtenir une seule valeur. L'idée derrière la modulation est la suivante : si les deux notes sont proches, alors le client et le fournisseur ont le même ressenti de la transaction ; ces notes reflètent vraisemblablement le comportement du fournisseur. Dans ce cas, la modulation récompense le fournisseur en augmentant légèrement la note résultante. Cependant, si les deux notes sont différentes, il y a deux possibilités : soit le fournisseur prétend qu'il s'est bien comporté alors que le client dit le contraire, soit le client médit sur le fournisseur, ce qui correspond au deuxième type de discriminations présenté en section 3.3.2. Ces deux possibilités étant indistinguables, nous diminuons la valeur de la note ; ainsi, si la note du client est bonne tandis que celle du fournisseur est élevée, la note résultante sera faible.

La modulation a donc trois paramètres : le seuil ℓ à partir duquel la note modulée est augmentée, la valeur m_+ de laquelle elle est augmentée lorsque les notes du client et du fournisseur sont égales, ainsi que la valeur m_- de laquelle elle est diminuée lorsque la différence est maximale. En notant $\tilde{\rho}$ la note modulée, on a $\tilde{\rho} = f_{\text{corr}}(\rho_{\text{nym}}, \rho_{\text{FS}})$ où f_{corr} est la fonction définie par :

$$
\begin{array}{rccl}
 & [0;1]\times[0;1] & \to & [0;1] \\
f_{\text{corr}} : & \rho_{\text{nym}}, \rho_{\text{FS}} & \mapsto & \frac{\rho_{\text{nym}}+\rho_{\text{FS}}}{2} + \begin{cases} \frac{m_+}{\ell}\cdot(\ell - |\rho_{\text{nym}} - \rho_{\text{FS}}|) & \text{si} \quad |\rho_{\text{nym}} - \rho_{\text{FS}}| \leqslant \ell, \\ \frac{m_-}{1-\ell}\cdot(|\rho_{\text{nym}} - \rho_{\text{FS}}| - \ell) & \text{si} \quad |\rho_{\text{nym}} - \rho_{\text{FS}}| \geqslant \ell. \end{cases}
\end{array}
$$

La figure 4.6 compare les résultats de cette fonction à la note du client pour $\ell = 0{,}1$, $m_+ = 0{,}05$, $m_- = -1$ et quatre valeurs de ρ_{nym} ; les traits fins représentent la note du client, tandis que les

traits épais représentent la note modulée. Par exemple, considérons que $\rho_{\text{nym}} = 0{,}33$, ce qui correspond aux courbes vertes de cette figure. La note modulée est supérieure à la note du client pour $\rho_{\text{FS}} \in [0{,}23; 0{,}51]$; celle-ci vaut au maximum $\rho_{\text{nym}} + m_+$, c'est-à-dire 0,38. Cependant, si la note du fournisseur est trop élevée – ou trop faible –, alors la note résultante devient relativement faible, et atteint même $\tilde{\rho} = 0{,}03$ quand $\rho_{\text{FS}} = 1$. Cette figure montre que le fournisseur n'a pas intérêt à donner une bonne note lorsqu'il sait que le client lui donne une mauvaise note. De plus, les légers biais que les clients peuvent exhiber sont adoucis : tant que les notes du client et du fournisseur sont proches, la note modulée est légèrement meilleure que la note du client, tandis qu'elle est moins bonne si le fournisseur ment sur son comportement.

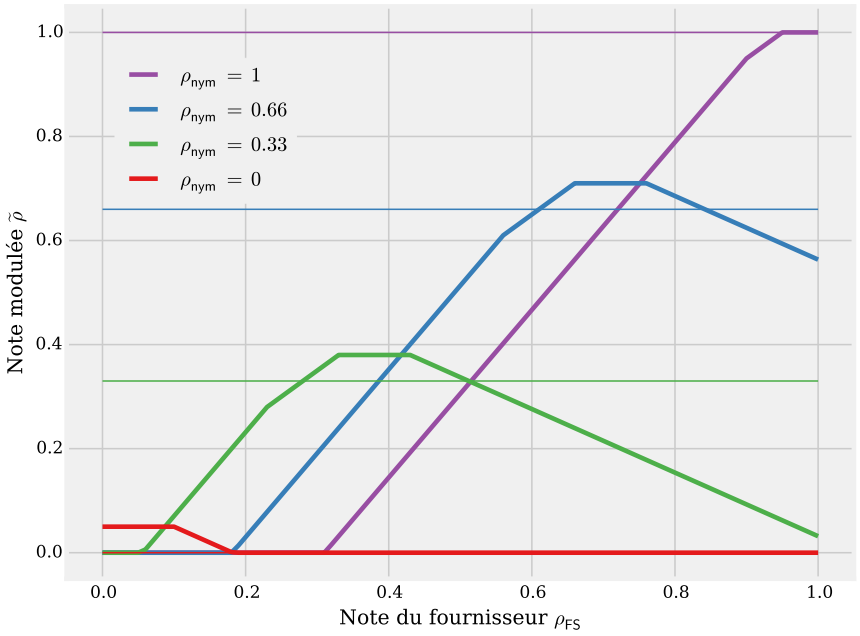


FIGURE 4.6 – Modulation des notes pour $\ell = 0{,}1$, $m_+ = 0{,}05$, et $m_- = -1$

Une fois qu'un témoignage a été modulé, il est valorisé à partir de la valeur v_ρ de la transaction associée, ce qui donne

$$\hat{\rho} = v_\rho \cdot \tilde{\rho}.$$

Cela permet de donner plus d'importance aux transactions de haute valeur. Par exemple, dans un système de commerce électronique, la valeur d'une transaction peut correspondre au prix payé par le client.

Finalement, un témoignage est vieilli. Jøsang et Ismail [JI02] proposent d'utiliser la fonction $t \mapsto \lambda^t$ à cet effet, où λ est un paramètre situé dans l'intervalle $[0; 1]$ et t est le temps écoulé depuis l'émission du témoignage. Par exemple, si le témoignage a été émis au temps t_ρ, sa valeur vieillie au temps t est $\bar{\rho} = \hat{\rho} \cdot \lambda^{t-t_\rho}$. Plus le paramètre λ est proche de 0, plus les témoignages récents sont importants.

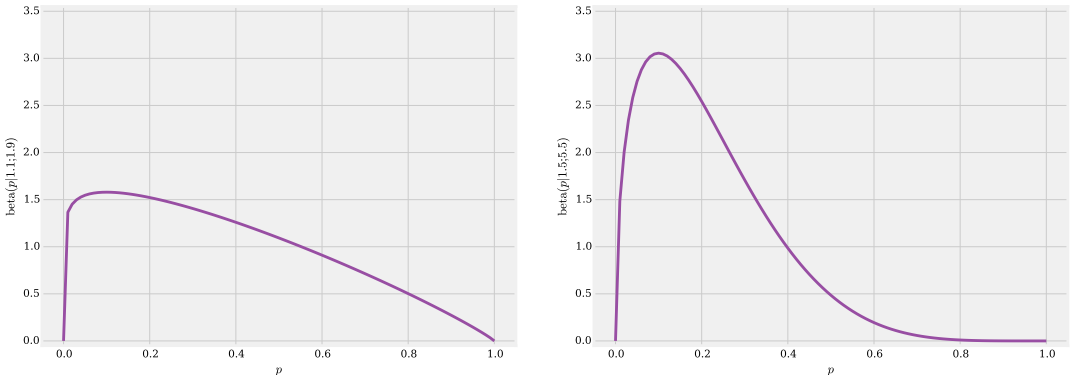
Ainsi, pour un historique de témoignages $T_{\text{FS}} = \left\{(\rho_{\text{nym}}, \rho_{\text{FS}}, v_\rho, t_\rho)\right\}$, la réputation obtenue

en utilisant la bêta réputation (voir section 3.2.2) est

$$\mathrm{rep}_{T_{\mathrm{FS}}}(t) = \frac{1 + \sum_{T_{\mathrm{FS}}} f_{\mathrm{corr}}(\rho_{\mathrm{nym}}, \rho_{\mathrm{FS}}) \cdot v_\rho \cdot \lambda^{t-t_\rho}}{2 + \sum_{T_{\mathrm{FS}}} v_\rho \cdot \lambda^{t-t_\rho}}.$$

Dans la suite, nous considérons que $v_\rho = 1$ – c'est à dire que toutes les transactions ont la même importance – par souci de simplification.

Comme expliqué précédemment (voir section 3.3.2), l'algorithme de filtrage proposé par Whitby et coll. [WJI04] compare les réputation locales – c'est-à-dire calculées à partir des témoignages émis par un unique client – à la réputation globale. Si, pour un client, les réputations locales et globales sont significativement différentes, alors ce client est considéré injuste et filtré. Dans notre cas, les clients sont anonymes et leurs pseudonymes ne sont pas associables. Les réputations locales ne peuvent donc regrouper les témoignages d'un seul pseudonyme ; dans le cas d'un anonymat idéal, un pseudonyme émet un unique témoignage. La figure 4.7a montre la loi bêta pour un unique témoignage négatif de 0,1 ; dans ce cas, on a $q_5 = 0{,}04$ et $q_{95} = 0{,}81$. Pour filtrer un témoignage, il faut que $\mathrm{rep}_{\mathrm{FS}} < q_5$ ou que $\mathrm{rep}_{\mathrm{FS}} > q_{95}$; un tel témoignage ne sera filtré que si la réputation du fournisseur est très bonne, ou exécrable (voir algorithme 3.1). Pour contourner ce problème, nous accentuons chaque témoignage d'un facteur f_F, c'est-à-dire que nous considérons que chaque pseudonyme a émis ses témoignages f_F fois. L'augmentation de ce facteur réduit les faux négatifs – c'est-à-dire les témoignages discriminatoires non filtrés – tout en augmentant les faux positifs – les témoignages légitimes filtrés. Ainsi, l'algorithme 3.1 est remplacé par l'algorithme 4.1. La figure 4.7b montre la loi bêta pour un même témoignage, mais accentué d'un facteur $f_F = 5$; dans ce cas, les quantiles sont $q_5 = 0{,}02$ et $q_{95} = 0{,}49$; ainsi, il est beaucoup plus probable qu'un tel témoignage soit filtré si le fournisseur a une bonne réputation.



(a) Sans facteur de filtrage

(b) Avec facteur de filtrage $f_F = 5$

FIGURE 4.7 – Loi bêta engendrée par un seul témoignage $\bar{\rho} = 0{,}1$

$F_{\mathsf{FS}} \leftarrow T_{\mathsf{FS}}$
repeat
  $\mathrm{rep}_{\mathsf{FS}} \leftarrow$ réputation calculée à partir de F_{FS}
  for all témoignage $\bar{\rho}$ de F_{FS} **do**
    $\mathrm{beta}_{\bar{\rho}} \leftarrow \mathrm{beta}\left(p|1 + f_F \cdot \bar{\rho}, 1 + f_F \cdot (1 - \bar{\rho})\right)$
    $q_5 \leftarrow$ quantile de 5 % de $\mathrm{beta}_{\bar{\rho}}$
    $q_{95} \leftarrow$ quantile de 95 % de $\mathrm{beta}_{\bar{\rho}}$
    if $q_5 > \mathrm{rep}_{\mathsf{FS}}$ ou $q_{95} < \mathrm{rep}_{\mathsf{FS}}$ **then**
      $F_{\mathsf{FS}} \leftarrow F_{\mathsf{FS}} \setminus \{\bar{\rho}\}$
until F_{FS} est stable

Listing 4.1 – Filtrage des témoignages injustes, adapté du filtrage de Whitby et coll. [WJI04]

4.5 Discussion

4.5.1 Précision théorique du score de réputation

Pour étudier la précision des scores de réputation, nous nous intéressons à la différence entre le score tel que calculé par un client, en présence de témoins médisants, et entre le comportement réel du fournisseur. Dans cette section, nous nous intéressons à la réputation calculée *avant* le filtrage, c'est-à-dire avec les témoignages T_{FS} (voir figure 4.5). Cette différence est maximale lorsque le comportement du fournisseur est extrême – par exemple très positif – et que les clients médisants émettent des notes à l'extrême opposé – dans le même exemple, des notes très négatives. Dans la suite, nous supposons que le fournisseur fournit un service correct, valant une note $\bar{\rho} = 1$, et que les notes des clients médisants sont $\bar{\rho} = 0$. Supposons que le fournisseur reçoive une note par intervalle de temps : la i-ième note est émise à $t = i$. Comme les notes sont vieillies, les notes reçues aux premiers rounds influent moins sur le score final que les dernières ; le pire cas se présente donc lorsque les témoignages médisants sont les derniers reçus. Ainsi, en notant n le nombre de témoignages reçus et m le nombre de témoignages médisants, le fournisseur a reçu $(n - m)$ témoignages tels que $\bar{\rho} = 1$, puis m témoignages tels que $\bar{\rho} = 0$. En considérant que chaque transaction a une valeur de 1 et que la réputation est calculée à $t = n + 1$, le comportement et la réputation sont donc définis par :

$$\mathrm{comp}_{\mathsf{FS}} = \frac{1 + \sum_{i=1}^{n} \lambda^{(n+1)-i}}{2 + \sum_{i=1}^{n} \lambda^{(n+1)-i}} \quad \text{et} \quad \mathrm{rep}_{\mathsf{FS}} = \frac{1 + \sum_{i=1}^{n-m} \lambda^{(n+1)-i}}{2 + \sum_{i=1}^{n} \lambda^{(n+1)-i}}$$

La différence relative entre les deux est donc

$$\delta_{\mathrm{rep}} = \left| \frac{\mathrm{comp}_{\mathsf{FS}} - \mathrm{rep}_{\mathsf{FS}}}{\mathrm{comp}_{\mathsf{FS}}} \right| \tag{4.1}$$

$$= \frac{\sum_{i=1}^{m} \lambda^{i}}{1 + \sum_{i=1}^{n} \lambda^{i}} \tag{4.2}$$

$$= \frac{\lambda - \lambda^{m+1}}{1 - \lambda^{n+1}}. \tag{4.3}$$

L'équation (4.2) montre que $\delta_{\text{rep}} \underset{\lambda \to 1}{\to} \frac{m}{n+1}$. En effet, quand $\lambda = 1$, les témoignages ne sont pas vieillis ; la différence entre le comportement du fournisseur et sa réputation est donc égale à la proportion de clients médisants. L'équation (4.3) montre de plus que $\delta_{\text{rep}} \underset{\lambda \to 0}{\to} 0$. À ce moment, le poids des anciens témoignages est nul, et le comportement et la réputation valent tous les deux 0,5. Cette équation montre également que, lorsque n et m deviennent très grand, la différence relative tend vers le facteur de vieillissement. Ces observations montrent que le choix du paramètre de vieillissement est crucial : s'il est trop proche de 0, la réputation de tous les fournisseurs sera proche de 0,5. Par contre, s'il est trop élevé, tous les témoignages auront la même importance, même les témoignages anciens. Dans la suite, nous utilisons $\lambda = 0{,}9$, ce qui est un compromis acceptable : les témoignages relativement récents sont encore importants, mais les âgés sont diminués ; par exemple, comme $0{,}9^{10} = 0{,}34$, un témoignage émis dix unités de temps plus tôt verra son importance diminuée de deux tiers.

4.5.2 Influence du filtrage sur la précision des scores de réputation

Tel que présenté, le filtrage des témoignages injustes dépend du facteur de filtrage f_F choisi. Nous nous intéressons donc à l'influence de ce paramètre sur le filtrage des témoignages. À cet effet, nous étudions la différence relative entre le score de réputation calculé par le client, rep_{FS}, et le comportement du fournisseur, comp_{FS} ; la différence relative δ_{rep} est définie par l'équation (4.1). Pour étudier cette différence, nous simulons des interactions entre un fournisseur de service et plusieurs clients.

Dans cette simulation, nous considérons un fournisseur de service qui est globalement honnête, mais qui échoue parfois à fournir un service correct. Ce fournisseur interagit avec des clients qui peuvent être honnêtes, ou malhonnêtes. Les clients honnêtes ajoutent un léger biais à leurs notes, qui exprime la subjectivité de leurs opinions ; ce biais est modélisé par une loi normale de moyenne nulle, quel que soit le comportement du fournisseur. De leur côté, les clients malhonnêtes émettent des notes qui sont tout le temps mauvaises, indépendamment du comportement du fournisseur. Symétriquement, nous aurions pu considérer un fournisseur se comportement globalement malhonnêtement, sauf avec un sous-ensemble favorisé de clients.

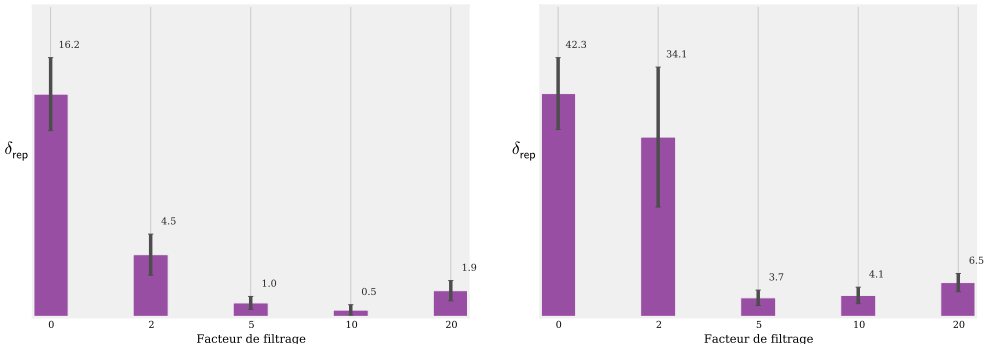
Plus précisément, le fournisseur se comporte honnêtement dans 95 % de ses transactions, et son comportement vaut alors une note distribuée uniformément dans l'intervalle $[0{,}8; 1]$. Le biais des clients honnêtes est modélisé par une distribution normale de moyenne 0 et d'écart-type 0,03, tandis que les notes des clients malhonnêtes – entre 15 % et 30 % des clients – sont distribuées uniformément dans l'intervalle $[0; 0{,}1]$.

Pour le calcul des scores de réputation, nous considérons les paramètres suivants :

$$\begin{aligned} m_+ &= 0{,}05, \\ m_- &= -1, \\ \ell &= 0{,}1, \\ \lambda &= 0{,}9. \end{aligned}$$

La simulation est divisée en 50 rounds. À chaque round, 10 clients sont aléatoirement choisis et interagissent avec le fournisseur. À la fin de la simulation, la différence relative δ_{rep} est calculée, pour différents paramètres de filtrage. La figure 4.8a présente la moyenne et l'écart-type de

δ_{rep}, en pourcentage, sur 100 simulations, pour plusieurs valeurs du facteur de filtrage – $f_F = 0$ revient à ne pas filtrer les témoignages, et sert donc de cas de base.



(a) Pour 15 % de clients malhonnêtes

(b) Pour 30 % de clients malhonnêtes

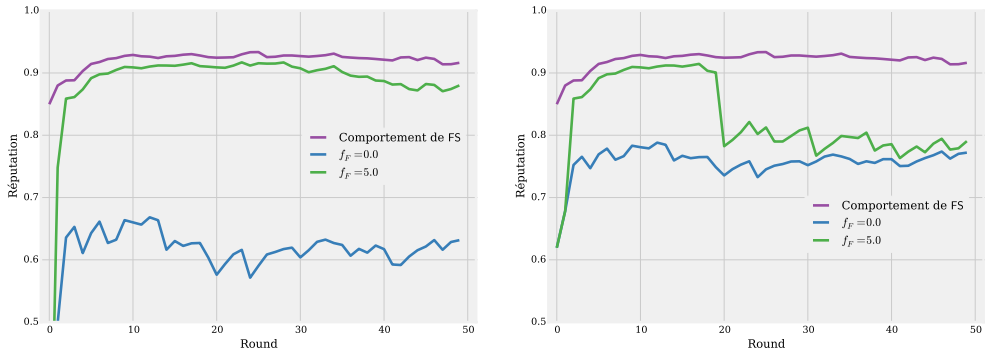
FIGURE 4.8 – Moyenne et écart-type de δ_{rep} sur 100 simulations

Comme nous pouvons le constater, sans filtrage, il y a une différence de 16,2 % entre la réputation calculée et le comportement réel du fournisseur, ce qui correspond aux 15 % de clients malhonnêtes. Cependant, dès que les témoignages sont filtrés – même avec un facteur de filtrage faible –, la différence est largement réduite : $\delta_{\text{rep}} = 4{,}5\,\%$ pour $f_F = 2$, et même $\delta_{\text{rep}} = 0{,}5\,\%$ pour $f_F = 10$. Cependant, si le facteur de filtrage est trop élevé, la précision du score de réputation diminue. Ce paramètre doit donc être choisi avec précaution. Notons que, tant que les clients médisants sont minoritaires, le filtrage des témoignages permet d'améliorer la précision du score de réputation. Par exemple, pour 30 % de clients malhonnêtes, la figure 4.8b montre que le filtrage des témoignages avec $f_F = 5$ permet de passer de $\delta_{\text{rep}} = 42{,}3\,\%$ à $\delta_{\text{rep}} = 3{,}7\,\%$.

La figure 4.9a présente l'évolution du comportement d'un fournisseur de service et de sa réputation pendant les 50 rounds de la simulation, avec et sans filtrage des témoignages. Cette figure montre deux choses : (a) le filtrage fonctionne dès les premiers rounds, c'est-à-dire même lorsque peu de témoignages sont disponibles, et (b) même sans être capable d'associer les témoignages des clients (propriété 7), le filtrage fonctionne.

Cependant, des clients malhonnêtes intelligents peuvent essayer de passer outre le filtrage des témoignages en augmentant leurs notes. La figure 4.9b présente l'évolution du comportement d'un fournisseur en présence de tels clients ; ces clients ne donnent plus une note uniformément distribuée dans l'intervalle $[0; 0{,}1]$ mais dans l'intervalle $[0{,}5; 0{,}6]$. Dans ce cas, dès qu'il y a suffisamment de témoignages médisants – après le vingtième round –, le score de réputation du fournisseur après filtrage est égal au score de réputation du fournisseur non filtré.

Un autre problème se pose pour les fournisseurs discriminant une partie de leurs clients. En effet, un fournisseur se comportant généralement correctement, mais se comportant mal avec 10 % de ses clients est indistinguable d'un fournisseur se comportant correctement, mais discriminé par 10 % de ses clients. Dans les deux cas, les témoignages négatifs sont filtrés ; ainsi, la réputation du fournisseur est bien meilleure que le comportement du fournisseur dans le premier cas.



(a) $0 \leqslant \bar{\rho} \leqslant 0{,}1$ (b) $0{,}5 \leqslant \bar{\rho} \leqslant 0{,}6$

FIGURE 4.9 – Évolution de la réputation d'un fournisseur pour 30 % de clients malhonnêtes

Une solution à ce problème est d'utiliser l'algorithme proposé par Whitby et coll. [WJI04], sans le facteur de filtrage. Dans ce cas, un client émettant plusieurs témoignages négatifs sera filtré, tandis qu'un client émettant un unique témoignage négatif ne le sera pas. Pour pouvoir utiliser cet algorithme, il est cependant nécessaire de garantir l'associabilité des témoignages pour savoir si le même client a émis deux témoignages (propriété 7).

4.5.3 Anonymat

Dans ce mécanisme de réputation, les clients génèrent eux-mêmes leurs pseudonymes de manière aléatoire et les enregistrent ensuite auprès de l'autorité centrale C. Comme un réseau de communication anonyme est utilisé, même l'autorité centrale est incapable d'associer les pseudonymes. Cependant, comme les fournisseurs ne sont pas anonymes, cette génération de pseudonymes permet à un utilisateur malveillant d'effectuer des attaques Sybil (voir section 3.3.4) ainsi que du bourrage d'urne (voir section 3.3.3).

Pour éviter ce problème, nous avons expliqué que l'autorité centrale demande aux utilisateurs de s'acquitter d'un coût monétaire ou calculatoire avant d'enregistrer un pseudonyme (voir section 4.3.1). Cependant, ce coût défavorise les petits clients qui ne peuvent effectuer trop de transactions sous peine d'être identifiables, tandis qu'il favorise les utilisateurs puissants qui peuvent mener facilement des attaques par bourrage d'urne sur leurs concurrents.

Une solution idéale à ce problème est de garantir l'associabilité des témoignages (propriété 7). Dans ce cas là, il est possible de repérer un utilisateur effectuant du bourrage d'urne sur un fournisseur ciblé, et de ne prendre en compte qu'un nombre limité de ces témoignages pour limiter l'attaque. Quand cette propriété est garantie, il n'est donc plus nécessaire de demander aux utilisateurs de s'acquitter d'un coût pour enregistrer un pseudonyme.

L'inconvénient majeur de ce mécanisme est qu'il ne préserve pas la vie privée des fournisseurs de service. De plus, les gestionnaires de score ne peuvent être mis en place que parce que les fournisseurs de service sont identifiés : un fournisseur est défini par les gestionnaires de score qui maintiennent ses témoignages. C'est-à-dire que la sécurité de ce mécanisme – aussi

bien en terme d'indéniabilité que d'inforgeabilité (propriétés 3 à 6) – repose sur l'identification des fournisseurs.

4.5.4 Bilan

Dans ce chapitre, nous avons présenté un mécanisme de réputation ainsi qu'un moteur de réputation. Nous avons montré que ce moteur de réputation est précis, et permet de filtrer les témoignages injustes. Nous avons malgré tout expliqué que la non-garantie de l'associabilité des témoignages limite l'utilité du filtrage dans certains cas.

Le mécanisme de réputation préserve la vie privée des clients en leur permettant de générer leurs propres pseudonymes. Cependant, afin d'éviter les attaques Sybil et les bourrages d'urne, la génération d'un pseudonyme doit coûter du temps ou de l'argent, ce qui favorise les utilisateurs puissants aux dépens des petits. L'associabilité des témoignages permet déjà d'empêcher les bourrages d'urne ; un mécanisme garantissant cette propriété n'a donc pas besoin de demander aux utilisateurs de payer un coût lorsqu'ils enregistrent leurs pseudonymes, ce qui rétablit l'équilibre entre les petits utilisateurs et les puissants. Nous proposons au chapitre suivant de nouveaux mécanismes de réputation, garantissant l'associabilité des témoignages et permettant aux clients de générer eux-mêmes leurs pseudonymes, sans qu'ils ne doivent s'acquitter d'un coût.

Outre ce déséquilibre, l'inconvénient majeur de ce mécanisme de réputation concerne l'identification des fournisseurs de service. De plus, c'est cette identification qui permet au mécanisme de garantir les propriétés de sécurité. Il paraît donc difficile de modifier légèrement ce mécanisme pour garantir à la fois l'anonymat des clients et celui des fournisseurs de service. Dans le chapitre suivant, nous présentons les modifications permettant de garantir simultanément l'anonymat des clients et des fournisseurs.

5 Mécanismes de réputation distribués, efficaces et préservant la vie privée

Ce chapitre présente la contribution principale de cette thèse : la construction d'un mécanisme de réputation préservant la vie privée des clients et des fournisseurs, et permettant aux clients d'émettre des témoignages positifs et négatifs, c'est-à-dire un mécanisme garantissant toutes les propriétés énoncées au chapitre 1 ; nous montrons de plus au chapitre 7 que cette construction est efficace.

Dans ce chapitre, nous construisons graduellement un tel mécanisme afin de montrer la réflexion qui a débouché sur ce mécanisme. Bien que cette approche en dilue les différentes avancées, nous pensons qu'elle permet de mieux l'appréhender, notamment en montrant les restrictions imposées par l'ajout progressif des propriétés de sécurité et de vie privée.

Dans un premier temps, nous présentons trois nouveaux outils cryptographiques : le calcul d'un invariant, qui concilie associabilité des témoignages (propriété 7) et respect de la vie privée des clients et des fournisseurs (propriétés 1 et 2) ; une instanciation d'un schéma de partage de secret vérifiable [Fel87], qui permet de prouver que le secret a été partagé correctement ; une variante *optimiste* de ce schéma, qui fait appel aux porteurs de part uniquement pour la reconstruction du secret, et non pas pour son partage.

À l'aide de ces nouveaux outils, nous modifions le mécanisme de réputation présenté au chapitre précédent pour garantir l'associabilité des témoignages et permettre aux clients de générer eux-mêmes leurs pseudonymes, sans faire appel à une autorité centrale. Nous présentons deux versions de ce mécanisme : la première utilise du partage de secret vérifiable classique, et la seconde sa variante optimiste.

Améliorer les deux variantes pour y intégrer le respect de la vie privée des fournisseurs n'est pas évident. Nous expliquons que pour y parvenir, nous devons utiliser des tierces parties, aussi bien pour la gestion des scores de réputation que pour la garantie de l'indéniabilité des témoignages. Nous motivons ces tierces parties et présentons leurs caractéristiques, puis expliquons comment les instancier et les choisir.

Finalement, armés de ces tierces parties, nous construisons le mécanisme de réputation recherché. Là encore, nous en présentons deux variantes : une reposant sur du partage de secret vérifiable classique, et une optimiste.

Les principes des quatre mécanismes proposés sont similaires ; une interaction se déroule en quatre étapes :

1. le client vérifie la réputation du fournisseur ;
2. le client et le fournisseur s'authentifient mutuellement ;
3. le client et le fournisseur se garantissent mutuellement l'indéniabilité des témoignages (les propriétés 3 et 4) ;

4. après la transaction, les deux partenaires peuvent émettre un témoignage.

Les deux premières étapes ont généralement lieu en même temps.

Plus précisément, les fournisseurs déposent leurs annonces de service sur un panneau d'affichage public ; ces annonces sont liées à leur identifiant ou, lorsque leur vie privée est préservée, à un de leurs pseudonymes. Un client intéressé par une annonce particulière peut vérifier la réputation du fournisseur lié ; cette vérification se fait soit auprès des gestionnaires de score du fournisseur, soit auprès du fournisseur lui-même, qui a préalablement reçu des signatures attestant sa réputation. Pour préserver sa vie privée, le client interagit sous un pseudonyme qu'il a généré lui-même et qui est utilisé uniquement pour cette interaction ; il prouve la validité de ce pseudonyme grâce à une preuve NIZK.

Une fois que le client a vérifié la réputation du fournisseur de service, et s'il est toujours intéressé, le client et le fournisseur se garantissent mutuellement l'indéniabilité des témoignages. Lorsque le fournisseur n'est pas anonyme, sa garantie est une simple signature ; les autres garanties reposent sur un des schémas de partage de secret vérifiable présentés en sections 5.1.2 et 5.1.3.

Le client et le fournisseur peuvent alors effectuer la transaction, quelle que soit sa nature. Ils peuvent ensuite émettre le témoignage de plusieurs manières différentes : lorsque le fournisseur n'est pas anonyme, le client est capable de l'émettre seul ; si le client refuse de le faire, le fournisseur peut demander l'assistance de ses gestionnaires de score. Lorsque le fournisseur est anonyme, le client et le fournisseur doivent collaborer pour en émettre un ; si l'un des deux refuse de participer, son partenaire peut malgré tout en émettre un grâce à la tierce partie introduite en section 5.3.2.

Finalement, les scores de réputation des fournisseurs de service sont mis à jour après une phase de synchronisation ; cette étape est différente lorsque la vie privée des fournisseurs est préservée.

5.1 Nouveaux outils cryptographiques

Avant de décrire les mécanismes de réputation préservant la vie privée des clients et des fournisseurs de service, nous devons enrichir la bibliothèque d'outils cryptographiques débutée au chapitre 2 de deux nouvelles constructions : l'invariant, qui permet de combiner associabilité et vie privée (propriétés 1, 2 et 7), ainsi qu'une instanciation d'un schéma de partage de secret vérifiable grâce au système de preuves de Groth et Sahai [GS08] ; nous en présentons également une version dite *optimiste*, qui ne requiert de tierce partie que lorsque la reconstruction du secret est nécessaire. Dans cette section, nous réutilisons les notations présentées au chapitre 2, et plus particulièrement en section 2.1.1 : nous considérons un groupe bilinéaire $\Lambda = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, G_1, G_2)$, ainsi qu'un élément de groupe $Y_1 \in \mathbb{G}_1$, paramètre public.

5.1.1 Calcul de l'invariant

Le but de l'invariant est de détecter les témoignages multiples d'un même client sur un même fournisseur de service pour empêcher les attaques par bourrage d'urne ; sans révéler l'identité des clients ; c'est-à-dire, de garantir à la fois l'associabilité des témoignages (propriété 7) et la

vie privée des clients (propriété 2). Cette approche est également proposée par Bethencourt et coll. [BSS10] ; néanmoins, le calcul que nous proposons ici est simplifié.

L'invariant est construit à partir des identifiants du client, $\mathsf{id}_{\mathsf{Cl}} \in \mathbb{Z}_p$, et du fournisseur, $\mathsf{Id}_{\mathsf{FS}} \in \mathbb{G}_1$; l'invariant est défini comme

$$\mathrm{Inv}(\mathsf{Id}_{\mathsf{FS}}, \mathsf{id}_{\mathsf{Cl}}) = (\mathsf{Id}_{\mathsf{FS}})^{\mathsf{id}_{\mathsf{Cl}}}.$$

Considérons un fournisseur FS et deux clients Cl_1 et Cl_2 ; les invariants associés sont égaux si et seulement si les clients sont également identiques, c'est-à-dire que

$$\mathrm{Inv}(\mathsf{Id}_{\mathsf{FS}}, \mathsf{id}_{\mathsf{Cl}_1}) = \mathrm{Inv}(\mathsf{Id}_{\mathsf{FS}}, \mathsf{id}_{\mathsf{Cl}_2}) \Leftrightarrow \mathsf{Cl}_1 = \mathsf{Cl}_2.$$

Ainsi, lors du calcul du score de réputation d'un fournisseur, les témoignages issus d'un bourrage d'urne comportent tous le même invariant ; pour empêcher une telle attaque, il est par exemple possible de ne prendre en compte que le dernier témoignage de chaque invariant.

De plus, en considérant deux fournisseurs distincts FS_1 et FS_2 ainsi que deux clients Cl_1 et Cl_2, la connaissance de $\mathrm{Inv}(\mathsf{Id}_{\mathsf{FS}_1}, \mathsf{id}_{\mathsf{Cl}_1})$ et $\mathrm{Inv}(\mathsf{Id}_{\mathsf{FS}_2}, \mathsf{id}_{\mathsf{Cl}_2})$ ne permet pas de savoir si $\mathsf{Cl}_1 = \mathsf{Cl}_2$. Ce résultat, prouvé en section 6.1.2, découle de l'hypothèse DDH.

L'invariant est un élément essentiel des témoignages ; aussi bien les clients que les fournisseurs doivent être capables de le construire (propriétés 3 et 4). De plus, lorsque la vie privée du fournisseur est préservée, le client ne doit pas découvrir $\mathsf{Id}_{\mathsf{FS}}$ avant d'avoir choisi sa note (propriété 1), et le fournisseur ne doit ni apprendre $\mathsf{id}_{\mathsf{Cl}}$, ni découvrir l'invariant avant la transaction (propriété 2). Pour garantir ces quatre propriétés, nous proposons de calculer l'invariant en trois étapes :

1. le fournisseur masque son identifiant en calculant un pré-invariant pre_inv, qu'il transmet au client ;
2. le client injecte son identifiant dans pre_inv et calcule un invariant masqué masked_inv ;
3. le client transmet masked_inv au fournisseur ; celui-ci « ouvre » masked_inv et obtient l'invariant inv.

Les deux premières étapes ne compromettent ni la vie privée du fournisseur, ni celle du client ; elles peuvent donc avoir lieu avant la transaction. La troisième n'a lieu qu'après le choix de sa

note par le client. Ces trois éléments sont définis par [1]

$$\begin{aligned}\mathsf{pre\_inv} &= \mathrm{Pre\_inv}(\mathsf{Id_{FS}}, r)\\ &= ({G_1}^r, \mathsf{Id_{FS}} \cdot {Y_1}^r)\,,\end{aligned}$$

$$\begin{aligned}\mathsf{masked\_inv} &= \mathrm{Mask}(\mathsf{pre\_inv}, \mathsf{id_{Cl}}, s)\\ &= \left({G_1}^s \cdot {Y_1}^{\mathsf{id_{Cl}}}, \mathsf{pre\_inv_1}^s \cdot \mathsf{pre\_inv_2}^{\mathsf{id_{Cl}}}\right)\\ &= \left({G_1}^s \cdot {Y_1}^{\mathsf{id_{Cl}}}, \mathsf{Id_{FS}}^{\mathsf{id_{Cl}}} \cdot ({G_1}^s \cdot {Y_1}^{\mathsf{id_{Cl}}})^r\right),\end{aligned}$$

$$\begin{aligned}\mathsf{inv} &= \mathrm{Unmask}(\mathsf{masked\_inv}, r)\\ &= \mathsf{masked\_inv_2} \cdot \mathsf{masked\_inv_1}^{-r}\\ &= (\mathsf{Id_{FS}})^{\mathsf{id_{Cl}}} = \mathrm{Inv}(\mathsf{Id_{FS}}, \mathsf{id_{Cl}}),\end{aligned}$$

où r et s sont des éléments choisis aléatoirement dans $\mathbb{Z}_p$, respectivement par le fournisseur et par le client, et où G_1 et Y_1 sont des éléments de $\mathbb{G}_1$ qui sont des paramètres publics.

Notons bien que le pré-invariant ne permet pas au client de déduire $\mathsf{Id_{FS}}$, ce qui préserve la vie privée du fournisseur.

5.1.2 Partage de secret vérifiable

Nous avons présenté en section 2.3.2 un schéma de partage de secret permettant de partager des éléments de groupe. À partir de ce schéma et d'un système de preuves NIZK (voir section 2.5) compatible avec les équations de groupe, nous construisons un schéma de partage de secret *vérifiable*. En pratique, nous instancions un tel schéma avec le système de preuves de connaissance de Groth et Sahai [GS08].

Dans cette section, nous supposons que chaque porteur de part possède une paire de clés de chiffrement asymétrique $(\mathsf{ek_{PP}}_i, \mathsf{dk_{PP}}_i)$; ils utilisent par exemple un schéma de chiffrement Elgamal [Elg85].

Rappelons que ce schéma de partage de secret (t, n) permet à un prouveur $\mathcal{P}$ de partager un secret $S \in \mathbb{G}_1$ en n parts $(i, S_i)_{1 \leqslant i \leqslant n}$ en choisissant aléatoirement un polynôme $Q : z \mapsto S \cdot \prod_{j=1}^{t-1} {A_j}^{(z^j)}$; les parts sont les $S_i = Q(i)$ pour $1 \leqslant i \leqslant n$.

Dans le cas du partage de secret vérifiable, le prouveur veut en plus convaincre un vérifieur $\mathcal{V}$ qu'un secret a effectivement été partagé, et pourra être reconstruit. Prouver que le partage est correct revient à prouver que :

- le secret est bien le secret attendu ;
- les parts sont générées à partir du même polynôme, qui vaut bien le secret en 0 ; c'est-à-dire que la reconstruction du secret à partir de t parts distinctes donnera le secret, pour tout ensemble de t parts distinctes.

1. Nous avons expliqué en section 2.1.1 que nous notons $X_{\mathbf{k}}$ le k-ième élément du multiplet X.

La première preuve, que nous notons π_S, dépend du secret partagé. Par exemple, si $\mathcal{P}$ possède un certificat cert_S de C sur S, cette preuve peut être

$$\pi_S = \textsc{nizk}\left\{S, \mathsf{cert}_S : \mathsf{Verif}(\mathsf{cert}_S, S, \mathsf{vk}_C)\right\},$$

pour ne révéler ni le secret, ni son certificat. La deuxième preuve peut se décomposer en n preuves – une par part –, définies par

$$\pi_{S_i} = \textsc{nizk}\left\{S, A_1, \ldots, A_{t-1} : S_i = S \cdot \prod_{j=1}^{t-1} A_j{}^{(i^j)}\right\},$$

pour $1 \leqslant i \leqslant n$. Une telle preuve garantit que la part S_i est calculée à partir du polynôme dont les coefficients sont engagés dans $C_S, C_{A_1}, \ldots, C_{A_j}$. Chacune des preuves doit être construite avec les mêmes engagements sur les coefficients du polynôme Q, sans quoi $\mathcal{P}$ peut générer chaque part avec un polynôme différent, ce qui empêche de reconstruire le secret attendu.

Pour prouver le partage à $\mathcal{V}$, le prouveur lui envoie les engagements $C_S, C_{A_1}, \ldots, C_{A_{t-1}}$ ainsi que la preuve π_S et les éléments permettant de la vérifier, ce qui garantit que le secret partagé est le secret attendu et annonce le polynôme utilisé pour générer les parts, sans les révéler. À chaque récipiendaire des parts – que nous appelons *porteurs de part* dans la suite –, $\mathcal{P}$ envoie la part, chiffrée, ainsi que la preuve π_{S_i} et les éléments permettant de la vérifier. Le chiffrement de la part empêche un adversaire interceptant les communications d'apprendre les parts, et donc le secret. Si la preuve reçue est valide pour la part et les engagements reçus, le porteur de part envoie une confirmation à $\mathcal{V}$; cette confirmation est une signature classique σ_{PP_i} sur $H(i, C_S, C_{A_1}, \ldots, C_{A_{t-1}})$. Le vérifieur accepte la confirmation si la signature est valide pour les engagements précédemment reçus de $\mathcal{P}$, ce qui garantit que les parts sont bien issues du même polynôme. La figure 5.1 présente ces échanges.

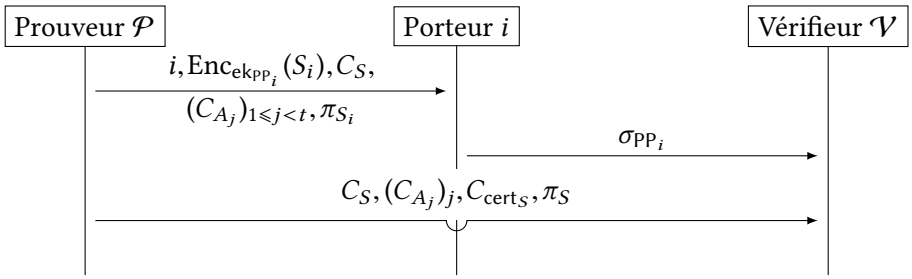


Figure 5.1 – Partage d'un secret

Si le vérifieur attend d'avoir reçu toutes les confirmations pour considérer le partage correct, un seul porteur de part malveillant peut bloquer la poursuite de l'interaction en n'envoyant pas sa confirmation. Mais si le seuil du partage est trop faible, une collusion de petite taille peut apprendre le secret. Pour déterminer le nombre de confirmations et le seuil du partage optimaux, notons ℓ le nombre de confirmations attendues par $\mathcal{V}$ et b le nombre de porteurs de part malveillants. Il faut alors que :

— le vérifieur puisse recevoir suffisamment de confirmations valides, c'est-à-dire $\ell \leqslant n - b$:

— les porteurs de part malveillants ne puissent pas reconstruire le secret, c'est-à-dire $b < t$;
— les porteurs de part honnêtes parmi ceux ayant confirmé leur part puissent reconstruire le secret, c'est-à-dire $t \leqslant \ell - b$.

En combinant ces équations, le choix optimal obtenu est

$$t = \lceil n/3 \rceil \quad \text{et} \quad \ell = 2t - 1,$$

ce qui tolère jusqu'à $b = \lceil n/3 \rceil - 1$ porteurs de part malveillants ; une collusion de cette taille ne peut ni bloquer l'interaction, ni reconstruire le secret.

Lorsque le vérifieur souhaite reconstruire le secret, les porteurs de part lui envoient la part S_i ainsi que la preuve π_{S_i} correspondante. Le vérifieur peut alors utiliser t parts valides pour reconstruire le secret attendu, et prouver la reconstruction avec les preuves π_{S_i} et π_S.

5.1.3 Partage de secret vérifiable optimiste

Nous expliquons en section 5.2.3 que le partage de secret vérifiable est utilisé pour reconstruire le secret uniquement dans le cas où le client est malveillant : si celui-ci est honnête, les parts ne sont pas utilisées. Cependant, même dans ce cas, les porteurs de part doivent participer au partage de secret vérifiable au début de l'interaction, au cas où l'un des deux utilisateurs se révèle être malveillant par la suite. Certains protocoles, dits *optimistes*, nécessitent une tierce partie uniquement lorsqu'un – ou plusieurs – des utilisateurs impliqués dans le protocole est malveillant [ASW97]. Dans cette section, nous présentons un partage de secret vérifiable *optimiste*, c'est-à-dire qui implique les porteurs de part uniquement lorsque le prouveur se révèle malveillant par la suite.

Similairement à la section précédente, nous considérons un secret $S \in \mathbb{G}_1$ sur lequel le prouveur possède un certificat cert_S émis par C. Les n parts sont les (i, S_i) générées à partir du polynôme $Q : z \mapsto S \cdot \prod_{j=1}^{t-1} A_j{}^{(z^j)}$.

Si le prouveur envoie directement les parts au vérifieur, celui-ci peut immédiatement reconstruire le secret. Pour éviter cela, $\mathcal{P}$ chiffre les parts pour les porteurs de part en calculant

$$E_{S_i} = \text{Enc}_{\text{ek}_{\text{PP}_i}}(S_i),$$

pour $1 \leqslant i \leqslant n$, et envoie les parts chiffrées au vérifieur. Le prouver doit alors garantir à $\mathcal{V}$ que :

— le secret est bien le secret attendu ;
— les parts sont générées à partir du même polynôme, qui vaut le secret en 0 ;
— les parts sont chiffrées avec les clés des porteurs de part correspondants ;
— les parts sont émises par le prouver.

Cette dernière garantie indique aux porteurs de part que les parts sont légitimes, et qu'ils peuvent donc les déchiffrer ; elle empêche le vérifieur de demander le déchiffrement d'une part arbitraire. Pour donner cette garantie, le prouveur signe chaque part :

$$\sigma_i = \text{SPSign}(\text{sk}_{\mathcal{P}}, S_i),$$

pour $1 \leqslant i \leqslant n$. Cependant, le prouver ne doit révéler ni la part, ni sa clé de signature, ni la signature elle-même. C'est pourquoi, comme expliqué en section 2.6.4, il utilise des preuves NIZK pour masquer ces éléments dans les équation de vérification du schéma de signature automorphe. Les preuves finales sont les

$$\begin{aligned}\pi_{E_{S_i}} = \text{NIZK}\Big\{ & \mathsf{vk}_{\mathcal{P}}, S_i, \sigma_i, S, A_1, \ldots, A_{t-1} : \\ & E_{S_i} = \text{Enc}_{\mathsf{ek}_{\mathsf{PP}_i}}(S_i) \\ & \wedge S_i = S \cdot \prod_{j=1}^{t-1} A_j{}^{(i^j)} \\ & \wedge \text{ SPVerif}(\sigma_i, S_i, \mathsf{vk}_{\mathcal{P}})\Big\},\end{aligned} \tag{5.1}$$

pour $1 \leqslant i \leqslant n$. Dans cette preuve, la première équation garantit le chiffrement de la part, la seconde garantit que toutes les parts sont issues du même polynôme, et la troisième authentifie la part.

Pour partager le secret, $\mathcal{P}$ envoie les parts chiffrées, les preuves ainsi que les engagements nécessaires à leur vérification à $\mathcal{V}$, c'est-à-dire E_{S_i}, $\pi_{E_{S_i}}$, C_{S_i} et C_{σ_i} pour $1 \leqslant i \leqslant n$, ainsi que C_S, C_{A_1}, …, $C_{A_{t-1}}$. Le vérifieur accepte le partage de secret si et seulement si les n preuves sont valides. À la différence du schéma de partage de secret non-optimiste, le vérifieur est alors convaincu que le prouveur a effectivement envoyé *toutes* les parts. Le nombre maximum de porteurs de part malveillants b ne dépend donc plus que de deux conditions : il faut que

— les porteurs de part malveillants ne puissent pas reconstruire le secret, c'est-à-dire $b < t$;

— les porteurs de part honnêtes puissent reconstruire le secret, c'est-à-dire $t \leqslant n - b$.

Le choix optimal obtenu est donc $t = \lceil n/2 \rceil$, ce qui tolère jusqu'à $b = \lceil n/2 \rceil - 1$ porteurs de part malveillants.

Pour reconstruire le secret, le vérifieur contacte les porteurs de part en leur envoyant la part chiffrée et la preuve associée, ainsi que tous les éléments permettant de la vérifier. Si cette preuve est valide, le porteur de part déchiffre la part et prouve ce déchiffrement avec une preuve NIZK π_{S_i} définie par

$$\begin{aligned}\pi_{S_i} = \text{NIZK}\Big\{ & \mathsf{dk}_{\mathsf{PP}_i} : \\ & S_i = \text{Dec}_{\mathsf{dk}_{\mathsf{PP}_i}}(E_{S_i}) \\ & \wedge \mathsf{dk}_{\mathsf{PP}_i} \text{ est la clé de déchiffrement associée à } \mathsf{ek}_{\mathsf{PP}_i}\Big\},\end{aligned}$$

et un engagement $C_{\mathsf{dk}_{\mathsf{PP}_i}}$ sur $\mathsf{dk}_{\mathsf{PP}_i}$. Le vérifieur pourra ensuite prouver la reconstruction du secret grâce aux preuves π_S, $\pi_{E_{S_i}}$ et π_{S_i}.

5.2 Retour sur le mécanisme préservant la vie privée des clients

Nous présentons dans cette section deux mécanismes de réputation dont les garanties sont similaires au mécanisme présenté lors du chapitre 4. Plus précisément, ces mécanismes préservent la vie privée des clients sans se préoccuper de celle des fournisseurs. Cependant, contrairement au mécanisme présenté au chapitre 4, ceux-ci permettent aux clients de générer autant

de pseudonymes que nécessaire sans communiquer avec l'autorité centrale et garantissent l'associabilité des témoignages, ce qui permet de mieux résister aux bourrages d'urne (propriété 7). Le premier de ces deux mécanismes utilise du partage de secret vérifiable classique, tandis que le second utilise du partage de secret vérifiable optimiste. Avant de détailler ces deux mécanismes, nous présentons la mise en place des utilisateurs, c'est-à-dire les éléments cryptographiques qu'ils utilisent.

5.2.1 Mise en place des utilisateurs

Nous considérons une autorité centrale C qui gère l'enregistrement des utilisateurs ; elle possède une paire de clés de signature automorphe $(\mathsf{sk}_C, \mathsf{vk}_C)$.

Similairement au mécanisme présenté en section 4.3, nous considérons trois types d'utilisateurs : des fournisseurs de service, des clients, ainsi que des gestionnaires de score. Les fournisseurs de service sont organisés sur une DHT, ce qui permet de déterminer aisément leurs gestionnaires de score. Nous considérons plus particulièrement un fournisseur de service FS, ses gestionnaires de score $(\mathsf{GS}_i)_{1 \leqslant i \leqslant n_{\mathsf{GS}}}$ et un client Cl.

Le fournisseur possède une paire de clés de signature classique (voir section 2.6.1) notée $(\mathsf{sk}_{\mathsf{FS}}, \mathsf{vk}_{\mathsf{FS}})$, un identifiant unique $\mathsf{Id}_{\mathsf{FS}}$ qui sert au calcul de l'invariant, ainsi qu'un certificat $\mathsf{cert}_{\mathsf{FS}}$ reçu de C portant sur $\langle \mathsf{vk}_{\mathsf{FS}}, \mathsf{Id}_{\mathsf{FS}} \rangle$.

Chaque gestionnaire de score possède deux paires de clés : une paire de clés de signature classique $(\mathsf{sk}_{\mathsf{GS}_i}, \mathsf{vk}_{\mathsf{GS}_i})$ et une paire de clés de chiffrement classique $(\mathsf{dk}_{\mathsf{GS}_i}, \mathsf{ek}_{\mathsf{GS}_i})$ – par exemple pour du chiffrement Elgamal [Elg85]. Les gestionnaires de score reçoivent également un certificat $\mathsf{cert}_{\mathsf{GS}_i}$ de C sur leurs clés publiques, ce qui leur permet de s'authentifier.

Finalement, le client possède une paire de clés de signature automorphe $(\mathsf{sk}_{\mathsf{Cl}}, \mathsf{vk}_{\mathsf{Cl}})$, un identifiant unique et secret $\mathsf{id}_{\mathsf{Cl}}$ qui sert au calcul de l'invariant, ainsi qu'un certificat $\mathsf{cert}_{\mathsf{Cl}}$ reçu de l'autorité d'enregistrement, c'est-à-dire une signature automorphe sur $\langle \mathsf{vk}_{\mathsf{Cl}}, \mathsf{id}_{\mathsf{Cl}} \rangle$. La propriété automorphe de cette signature est essentielle, puisqu'elle permet de prouver que le client est valide – c'est-à-dire qu'il possède une telle signature – sans révéler ni son identifiant, ni la signature elle-même (voir section 2.6.4). Le client génère lui-même ses pseudonymes. Pour se faire, il s'engage sur sa clé de vérification en calculant [2]

$$\mathsf{nym}_{\mathsf{Cl}} = \mathrm{Com}(\mathsf{vk}_{\mathsf{Cl}}, \_).$$

Pour s'authentifier, le client s'engage également sur son identifiant et sur son certificat :

$$C_{\mathsf{id}_{\mathsf{Cl}}} = \mathrm{Com}(\mathsf{id}_{\mathsf{Cl}}, \_)$$
$$C_{\mathsf{cert}_{\mathsf{Cl}}} = \mathrm{Com}(\mathsf{cert}_{\mathsf{Cl}}, \_).$$

À partir de ces engagements, le client calcule une preuve de validité de son certificat, c'est-à-

2. La notation « _ » signifie que l'aléa utilisé pour l'engagement n'est pas nécessaire à la compréhension du mécanisme (voir section 2.4.2).

dire une preuve NIZK $\pi_{\mathsf{cert}_{\mathsf{Cl}}}$ définie par :

$$\pi_{\mathsf{cert}_{\mathsf{Cl}}} = \text{NIZK}\Big\{\mathsf{cert}_{\mathsf{Cl}}, \mathsf{vk}_{\mathsf{Cl}}, \mathsf{id}_{\mathsf{Cl}} : \\ \mathsf{SPVerif}(\mathsf{cert}_{\mathsf{Cl}}, \langle \mathsf{vk}_{\mathsf{Cl}}, \mathsf{id}_{\mathsf{Cl}} \rangle, \mathsf{vk}_C) \\ \wedge e(V_{\mathsf{Cl}}, G_2) = e(G_1, W_{\mathsf{Cl}})\Big\}.$$

Comme expliqué en section 2.6.3, cette preuve permet au client de construire des signatures proxy anonymes. Ces engagements et cette preuve sont utilisés pour une interaction unique ; une fois celle-ci terminée, ils ne sont plus réutilisés.

5.2.2 Mécanisme préservant la vie privée des clients

L'interaction entre le client et le fournisseur suit le plan énoncé au début de ce chapitre : dans un premier temps, le client obtient la réputation du fournisseur auprès de ses gestionnaires de score. Ensuite, le client et le fournisseur s'authentifient mutuellement, et se garantissent l'indéniabilité de la note et de la preuve de transaction. Après ces étapes, la transaction entre le client et le fournisseur peut avoir lieu. Finalement, le client et le fournisseur émettent un témoignage ; cette dernière étape peut se dérouler différemment suivant le comportement du client.

Obtention de la réputation du fournisseur

Lorsque le client est intéressé par les services offerts par un fournisseur donné, il contacte ses gestionnaires de score au travers de la DHT. Les gestionnaires de score lui répondent avec les témoignages émis sur le fournisseur, qu'ils signent. Cette étape n'est pas modifiée par rapport au mécanisme précédent ; la section 4.3.2 la décrit.

Authentification mutuelle

Pour s'authentifier auprès du fournisseur, le client lui envoie $\mathsf{nym}_{\mathsf{Cl}}$, $C_{\mathsf{id}_{\mathsf{Cl}}}$, $C_{\mathsf{cert}_{\mathsf{Cl}}}$ ainsi que $\pi_{\mathsf{cert}_{\mathsf{Cl}}}$. De son côté, le fournisseur signe $\mathsf{nym}_{\mathsf{Cl}}$ en calculant

$$\sigma_{\mathsf{nym}} = \mathsf{Sign}(\mathsf{sk}_{\mathsf{FS}}, \mathsf{nym}_{\mathsf{Cl}}),$$

et envoie la signature accompagnée de $\mathsf{vk}_{\mathsf{FS}}$, $\mathsf{Id}_{\mathsf{FS}}$ et $\mathsf{cert}_{\mathsf{FS}}$ au client. Cette signature garantit qu'un client ne peut témoigner que sur les fournisseurs avec qui il va interagir, et permet au client d'émettre un témoignage après la transaction. La figure 5.2 décrit cet échange.

Partage de l'invariant

Une fois que les deux partenaires se sont authentifiés, le client calcule l'invariant et prouve son calcul en masquant l'invariant et son identifiant :

$$\mathsf{inv} = \mathsf{Inv}(\mathsf{Id}_{\mathsf{FS}}, \mathsf{id}_{\mathsf{Cl}})$$
$$\pi_{C_{\mathsf{inv}}} = \text{NIZK}\Big\{\mathsf{inv}, \mathsf{id}_{\mathsf{Cl}} : \mathsf{inv} = \mathsf{Inv}(\mathsf{Id}_{\mathsf{FS}}, \mathsf{id}_{\mathsf{Cl}})\Big\}.$$

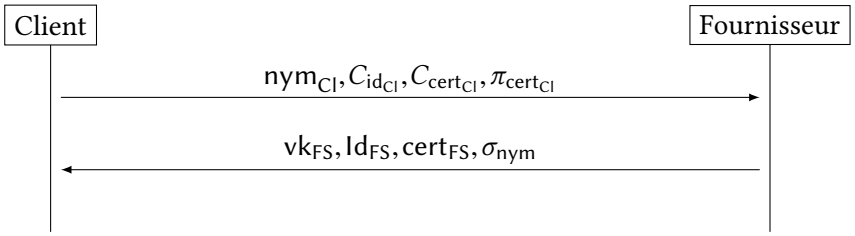


FIGURE 5.2 – Authentification mutuelle du client et du fournisseur

En effet, le fournisseur n'est pas anonyme ; le client a donc accès à $\mathsf{Id}_{\mathsf{FS}}$. Le client réutilise l'engagement $C_{\mathsf{id}_{\mathsf{Cl}}}$ sur $\mathsf{id}_{\mathsf{Cl}}$, qui a déjà été utilisé pour la preuve $\pi_{\mathsf{cert}_{\mathsf{Cl}}}$, et calcule un engagement C_{inv} sur inv. Le client partage ensuite l'invariant en utilisant les gestionnaires de score comme porteurs de part (voir section 5.1.2) : le secret est inv, sa preuve $\pi_{C_{\mathsf{inv}}}$, les coefficients du polynôme sont notés $A_1, \ldots, A_{t_{\mathsf{GS}}-1}$, et leurs engagements $C_{A_1}, \ldots, C_{A_{t_{\mathsf{GS}}-1}}$; les parts sont notées S_i et leurs preuves π_{S_i}, pour $1 \leqslant i \leqslant n_{\mathsf{GS}}$. Une fois ces éléments calculés, le client envoie les engagements et la preuve $\pi_{C_{\mathsf{inv}}}$ au fournisseur, et les engagements, $\mathrm{Enc}_{\mathsf{ek}_{\mathsf{GS}_i}}(S_i)$ et π_{S_i} au i-ième gestionnaire de score. Si la part reçue par un gestionnaire de score est valide, celui-ci envoie une signature σ_{GS_i} au fournisseur définie par

$$\sigma_{\mathsf{GS}_i} = \mathrm{Sign}\left(\mathsf{sk}_{\mathsf{GS}_i}, H(i, C_{\mathsf{inv}}, C_{A_1}, \ldots, C_{A_{t_{\mathsf{GS}}-1}})\right).$$

La réception d'une signature valide par le fournisseur signifie que le gestionnaire de score a reçu une part valide pour les engagements reçus par le fournisseur. Si le fournisseur reçoit $2\lceil n_{\mathsf{GS}}/3 \rceil - 1$ parts valides, alors $2\lceil n_{\mathsf{GS}}/3 \rceil - 1$ gestionnaires de score ont reçus des parts qui ont été générées à partir du secret attendu et d'un même polynôme ; nous avons expliqué en section 5.1.2 pourquoi le fournisseur attend ce nombre de parts. Dans ce cas, le fournisseur accepte le partage de secret. La figure 5.3 présente cette étape.

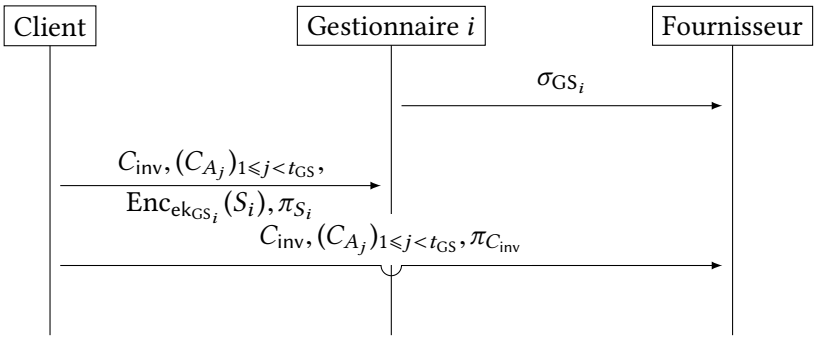


FIGURE 5.3 – Partage de l'invariant

Une fois le partage de l'invariant terminé, le client et le fournisseur peuvent effectuer la transaction.

Remarque (Émission des témoignages)
Les gestionnaires de score n'ont aucun moyen de vérifier que la transaction a effectivement eu lieu. Ainsi, ils acceptent un témoignage tant que celui-ci comprend la signature du fournisseur – qui signifie son accord – et l'invariant – qui donne l'accord du client. ■

Construction du témoignage

L'émission du témoignage peut se dérouler de deux manières : si le client veut noter le fournisseur, il est capable de construire le témoignage seul. Cependant, s'il ne le désire pas, alors le fournisseur peut utiliser les parts de ses gestionnaires de score pour obtenir un témoignage.

Scénario A – client honnête Si le client désire émettre un témoignage, il construit une preuve du calcul de l'invariant π_{inv} ; à la différence de $\pi_{C_{\mathsf{inv}}}$, cette preuve ne masque pas l'invariant :

$$\pi_{\mathsf{inv}} = \textsc{nizk}\Big\{\mathsf{id}_{\mathsf{Cl}} : \mathsf{inv} = \mathrm{Inv}(\mathsf{Id}_{\mathsf{FS}}, \mathsf{id}_{\mathsf{Cl}})\Big\}.$$

Cette preuve garantit que l'invariant donné par le client est calculé correctement. Le client choisit ensuite une note ρ sur laquelle il calcule une signature proxy anonyme ς_ρ :

$$\varsigma_\rho = \mathrm{APSign}\left(\mathsf{sk}_{\mathsf{Cl}}, H(\sigma_{\mathsf{nym}}, \mathsf{inv}, \rho)\right),$$

qui garantit que c'est bien le client qui choisit sa note. Cette signature est proxy anonyme pour éviter de révéler la clé de vérification du client, qui le désanonymiserait. Finalement, le client envoie les éléments suivants aux gestionnaires de score du fournisseur : $\mathsf{nym}_{\mathsf{Cl}}$, $C_{\mathsf{id}_{\mathsf{Cl}}}$, $C_{\mathsf{cert}_{\mathsf{Cl}}}$, $\pi_{\mathsf{cert}_{\mathsf{Cl}}}$, $\mathsf{Id}_{\mathsf{FS}}$, σ_{nym}, inv, π_{inv}, ρ, ς_ρ. Un gestionnaire de score accepte un tel témoignage si les deux preuves et les deux signatures incluses sont valides.

Le moteur de réputation présenté au chapitre 4 permet au fournisseur d'émettre une note qui sera ensuite modulée avec celle du client ; pour ajouter cette fonctionnalité au protocole, il suffit que le fournisseur envoie cette note ainsi que $\mathsf{nym}_{\mathsf{Cl}}$ aux gestionnaires de score. La première colonne de la table 5.1 résume les éléments composant le témoignage.

	Scénario A	Scénario B
Fournisseur	$\mathsf{vk}_{\mathsf{FS}}, \mathsf{Id}_{\mathsf{FS}}$	$\mathsf{vk}_{\mathsf{FS}}, \mathsf{Id}_{\mathsf{FS}}$
Client	$\mathsf{nym}_{\mathsf{Cl}}, C_{\mathsf{id}_{\mathsf{Cl}}}, C_{\mathsf{cert}_{\mathsf{Cl}}}, \pi_{\mathsf{cert}_{\mathsf{Cl}}}$	$\mathsf{nym}_{\mathsf{Cl}}, C_{\mathsf{id}_{\mathsf{Cl}}}, C_{\mathsf{cert}_{\mathsf{Cl}}}, \pi_{\mathsf{cert}_{\mathsf{Cl}}}$
Inforgeabilité	σ_{nym}	σ_{nym}
Invariant	$\mathsf{inv}, \pi_{\mathsf{inv}}$	$\mathsf{inv}, C_{\mathsf{inv}}, (C_{A_j})_{1 \leqslant j < t_{\mathsf{GS}}}, \pi_{C_{\mathsf{inv}}}, (S_{i_j}, \pi_{S_{i_j}})_{1 \leqslant j \leqslant t_{\mathsf{GS}}}$
Note	ρ, ς_ρ	N.A.

Table 5.1 – Éléments composant le témoignage

Scénario B – client malveillant Si au bout d'un certain temps le client n'a toujours pas émis de note, le fournisseur peut demander les parts conservées par ses gestionnaires de score.

Après un temps d'attente permettant potentiellement au client de témoigner, les gestionnaires de score renvoient S_i ainsi que π_{S_i} au fournisseur. Une fois que le fournisseur a reçu t_{GS} parts valides, il peut utiliser l'interpolation de Lagrange pour obtenir le secret, c'est-à-dire l'invariant (voir section 2.3.2). Le fournisseur peut alors obtenir un témoignage en envoyant les éléments suivants à ses gestionnaires de score : $\mathsf{Id}_{\mathsf{FS}}$, inv, $\mathsf{nym}_{\mathsf{Cl}}$, $C_{\mathsf{id}_{\mathsf{Cl}}}$, $C_{\mathsf{cert}_{\mathsf{Cl}}}$, C_{inv}, $(C_{A_j})_{1\leqslant j<t_{\mathsf{GS}}}$ σ_{nym}, $\pi_{\mathsf{cert}_{\mathsf{Cl}}}$, $\pi_{C_{\mathsf{inv}}}$, $(S_{i_j}, \pi_{S_{i_j}})_{1\leqslant j\leqslant t_{\mathsf{GS}}}$. Un gestionnaire de score accepte un tel témoignage si les $t_{\mathsf{GS}} + 2$ preuves et la signature sont valides. La figure 5.4 décrit cet échange.

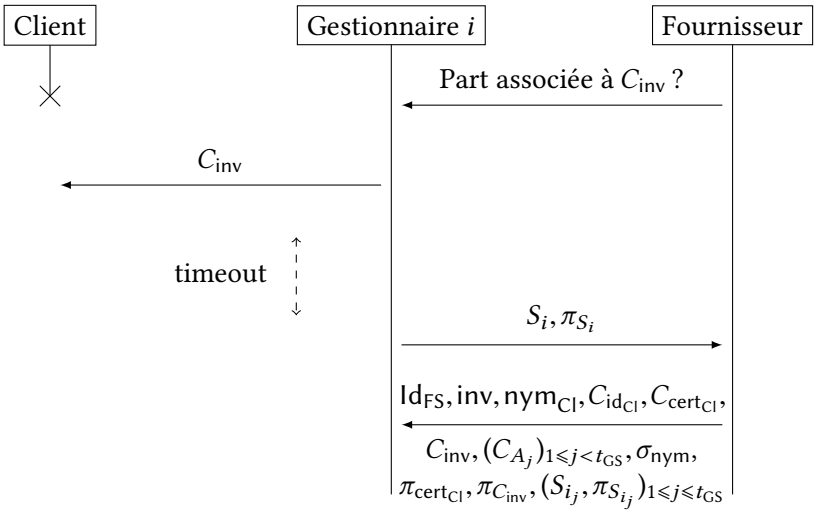


FIGURE 5.4 – Construction du témoignage lorsque le client est malveillant

Similairement au mécanisme précédent, les gestionnaires de score se synchronisent une fois que le témoignage résultant de la transaction a été émis (voir section 4.3.3) ; chacun propose une valeur, le témoignage, et ils utilisent un algorithme du consensus pour décider de la valeur à accepter. Les témoignages comportent toutes les informations prouvant leur émission, c'est-à-dire les éléments de la table 5.1. Les gestionnaires de score peuvent donc proposer soit un témoignage valide, proposé par le client, soit un témoignage invalide, détecté comme tel – ce qui revient à ne rien proposer ; l'algorithme du consensus utilisé ne doit donc pas nécessairement tolérer les comportements byzantins. Le protocole décrit par Raynal résolvant le problème du consensus uniforme pour des fautes par omission peut donc être utilisé [Ray02] ; ce protocole tolère jusqu'à $\lceil n_{\mathsf{GS}}/2 \rceil - 1$ utilisateurs fautifs. Néanmoins, le partage de secret vérifiable utilisé ne tolère que jusqu'à $\lceil n_{\mathsf{GS}}/3 \rceil - 1$ gestionnaires de score malveillants (voir section 5.1.2). Ainsi, ce mécanisme tolère jusqu'à $\lceil n_{\mathsf{GS}}/3 \rceil - 1$ gestionnaires malveillants, et il faut autant de gestionnaires de score par fournisseur que pour le mécanisme précédent, c'est-à-dire 103 (voir section 4.2).

Remarquons de plus que le fournisseur n'a rien à faire une fois qu'il a rendu son service ; il lui suffit de vérifier que le client a émis son témoignage et, si nécessaire, de faire appel aux gestionnaires de score pour obtenir une preuve de transaction.

5.2.3 Mécanisme optimiste préservant la vie privée des clients

Le mécanisme précédent préserve la vie privée des clients tout en garantissant l'indéniabilité et l'inforgeabilité des témoignages (voir section 5.2.2). Cependant, le partage de secret utilisé par ce mécanisme fait appel aux gestionnaires de score même dans le scénario A, c'est-à-dire lorsqu'ils ne sont pas nécessaires pour la construction du témoignage. Nous avons présenté en section 5.1.3 un schéma de partage de secret, dit optimiste, qui résout ce problème. Nous décrivons une variante du mécanisme de réputation utilisant ce schéma de partage optimiste, c'est-à-dire que cette variante ne fait appel aux gestionnaires de score que pour la construction du témoignage dans le scénario B, et pour son émission.

L'authentification mutuelle entre le client et le fournisseur, et la construction du témoignage dans le scénario A ne font pas appel aux gestionnaires de score ; ces deux étapes sont donc inchangées.

Dans ce mécanisme optimiste, la synchronisation des gestionnaires de score et le partage de secret vérifiable optimiste tolèrent jusqu'à $\lceil n_{GS}/2 \rceil - 1$ gestionnaires de score malveillants. La démarche suivie en section 4.2 peut être appliquée en remplaçant $f : n \mapsto \lceil n/3 \rceil - 1$ par $f : n \mapsto \lceil n/2 \rceil - 1$; avec les mêmes paramètres que précédemment, il faut dorénavant choisir $n_{GS} = 51$ gestionnaires de score.

Obtention de la réputation du fournisseur

Dans le mécanisme classique précédent (voir section 5.2.2), le client obtient les témoignages émis sur le fournisseur auprès des gestionnaires de score, ce qui lui permet de calculer la réputation du fournisseur. Dans le mécanisme optimiste, nous supposons qu'ils ne sont pas nécessairement en ligne ; l'obtention de la réputation du fournisseur de cette manière n'est donc plus possible. À la place, les gestionnaires de score calculent eux-mêmes la réputation du fournisseur, rep_{FS} et signent la réputation, l'identifiant du fournisseur et une estampille timestamp pour empêcher le fournisseur d'utiliser une ancienne réputation :

$$\sigma_{\mathsf{rep},GS_i} = \mathrm{Sign}\left(\mathsf{sk}_{GS_i}, H(\mathsf{Id}_{FS}, \mathsf{timestamp}, \mathsf{rep}_{FS})\right).$$

Ils envoient ensuite la réputation et la signature au fournisseur. Ainsi, un client demande directement sa réputation au fournisseur, qui la lui transmet accompagnée des signatures. Le client accepte la réputation si au moins une majorité $t_{GS} = \lceil n_{GS}/2 \rceil$ des signatures sont valides, et sont bien émises par les gestionnaires de score du fournisseur. La figure 5.5 décrit cet échange.

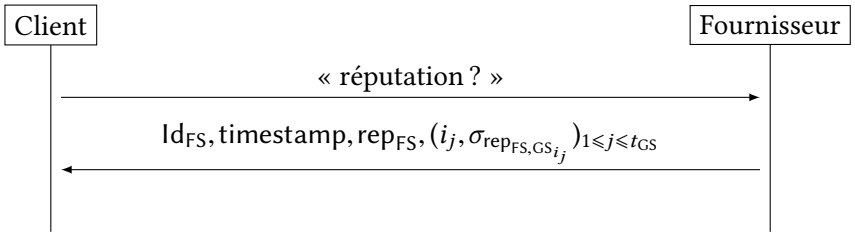


FIGURE 5.5 – Obtention de la réputation du fournisseur, pour la variante optimiste

Une fois que le client a vérifié la réputation du fournisseur, ils s'authentifient mutuellement, comme dans le mécanisme précédent.

Partage optimiste de l'invariant

Pour partager l'invariant, le client commence par le calculer et prouver ce calcul, comme précédemment :

$$\begin{aligned} \mathsf{inv} &= \mathrm{Inv}(\mathsf{Id}_{\mathsf{FS}}, \mathsf{id}_{\mathsf{Cl}}) \\ \pi_{C_{\mathsf{inv}}} &= \textsc{nizk}\left\{\mathsf{inv}, \mathsf{id}_{\mathsf{Cl}} : \mathsf{inv} = \mathrm{Inv}(\mathsf{Id}_{\mathsf{FS}}, \mathsf{id}_{\mathsf{Cl}})\right\}. \end{aligned}$$

Il chiffre ensuite les parts pour les gestionnaires de score :

$$E_{S_i} = \mathrm{Enc}_{\mathsf{ek}_{\mathsf{GS}_i}}(S_i),$$

pour $1 \leqslant i \leqslant n_{\mathsf{GS}}$, pour lesquelles il construit des signatures σ_i et des preuves $\pi_{E_{S_i}}$ (voir section 5.1.3). Le client envoie les parts chiffrées, les preuves ainsi que les engagements au fournisseur, qui accepte le partage si toutes les signatures et preuves sont valides. La figure 5.6 décrit cet envoi.

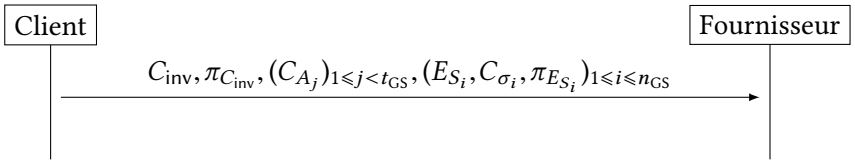


FIGURE 5.6 – Partage optimiste de l'invariant

Remarque
La preuve de réputation, l'authentification mutuelle du client et du fournisseur, ainsi que le partage de l'invariant peuvent être fusionnés pour réduire le nombre de messages échangés entre le client et le fournisseur. Nous n'avons pas présenté cette fusion pour séparer les objectifs de chaque étape. ■

Construction du témoignage lorsque le client est malveillant

Si au bout d'un certain temps le client n'a pas émis de témoignage, le fournisseur réclame les parts des gestionnaires de score. Pour ce faire, il envoie E_{S_i}, $\mathsf{nym}_{\mathsf{Cl}}$, C_{σ_i}, C_{inv}, $(C_{A_j})_j$ et $\pi_{E_{S_i}}$ à son i-ième gestionnaire de score. Le gestionnaire de score vérifie la preuve et, si elle est valide, déchiffre la part :

$$S_i = \mathrm{Dec}_{\mathsf{dk}_{\mathsf{GS}_i}}(E_{S_i}).$$

Il construit également la preuve π_{S_i} décrite en section 5.1.3, grâce à un engagement $C_{\mathsf{dk}_{\mathsf{GS}_i}}$ sur sa clé de déchiffrement. Le gestionnaire de score renvoie finalement S_i, $C_{\mathsf{dk}_{\mathsf{GS}_i}}$ et π_{S_i} au fournisseur. Celui-ci peut alors reconstruire l'invariant, qui sert de preuve de transaction, et émettre un témoignage en envoyant les éléments présentés à la deuxième colonne de la table 5.2 aux gestionnaires de score ; ce témoignage ne comporte pas de note du client.

	Scénario A	Scénario B
Fournisseur	$\mathsf{Id}_{\mathsf{FS}}, \mathsf{vk}_{\mathsf{FS}}$	$\mathsf{Id}_{\mathsf{FS}}, \mathsf{vk}_{\mathsf{FS}}$
Client	$\mathsf{nym}_{\mathsf{Cl}}, C_{\mathsf{id}_{\mathsf{Cl}}}, C_{\mathsf{cert}_{\mathsf{Cl}}}, \pi_{\mathsf{cert}_{\mathsf{Cl}}}$	$\mathsf{nym}_{\mathsf{Cl}}, C_{\mathsf{id}_{\mathsf{Cl}}}, C_{\mathsf{cert}_{\mathsf{Cl}}}, \pi_{\mathsf{cert}_{\mathsf{Cl}}}$
Inforgeabilité	σ_{nym}	σ_{nym}
Invariant	$\mathsf{inv}, \pi_{\mathsf{inv}}$	$\mathsf{inv}, C_{\mathsf{inv}}, (C_{A_j})_{1 \leqslant j < t_{\mathsf{GS}}}, \pi_{C_{\mathsf{inv}}}, (E_{S_{i_j}}, S_{i_j}, C_{\sigma_{i_j}}, C_{\mathsf{dk}_{\mathsf{GS}_{i_j}}}, \pi_{E_{S_{i_j}}}, \pi_{S_{i_j}})_{1 \leqslant j \leqslant t_{\mathsf{GS}}}$
Note	ρ, ς_ρ	N.A.

Table 5.2 – Éléments composant le témoignage pour la variante optimiste

5.3 Tierces parties distribuées et vie privée des fournisseurs

Les mécanismes de réputation doivent garantir que l'émission des témoignages – positifs *et* négatifs – ainsi que leur agrégation – en vue du calcul des scores de réputation – se déroulent correctement. Confier ces deux opérations aux fournisseurs eux-mêmes paraît complexe et coûteux : il faudrait alors garantir que les scores de réputation ne sont pas modifiés et qu'aucun témoignage, même négatif, ne soit « accidentellement » oubliés.

Nous avons montré aussi bien au chapitre 4 qu'en section 5.2 qu'une tierce partie distribuée, les gestionnaires de score, est capable de garantir le bon déroulement de ces deux opérations – c'est-à-dire, garantir les propriétés de sécurité présentées en section 1.4. Cette tierce partie garantit à la fois l'intégrité des témoignages et des scores de réputation, ainsi que l'indéniabilité des témoignages : elle stocke les témoignages sur un fournisseur particulier, et permet à la fois aux clients et aux fournisseurs de témoigner lorsqu'ils sont en mesure de prouver leur participation à une transaction.

Cependant, plusieurs problèmes se posent lorsque la vie privée des fournisseurs doit être préservée et qu'ils utilisent des pseudonymes, et non plus leur identifiant unique, pour interagir. Tout d'abord, les clients ne doivent plus avoir accès aux témoignages : les témoignages, même lorsqu'ils concernent un pseudonyme, donnent aux clients plus d'informations que la seule réputation, ce qui réduit l'ensemble d'anonymat des fournisseurs. Ensuite, il n'est plus possible d'avoir un ensemble de gestionnaires de score par fournisseur : deux pseudonymes présentant les mêmes gestionnaires de score sont probablement issus du même fournisseur.

Dans cette section, nous séparons les deux rôles précédemment remplis par les gestionnaires de score et listons les fonctionnalités requises par ces deux rôles. Nous motivons ensuite l'utilisation de deux types de tierces parties différentes pour remplir ces deux rôles, et expliquons comment les choisir.

5.3.1 Gestion des scores de réputation

Certains mécanismes de réputation assignent une tierce partie à la gestion des scores d'un fournisseur donné [Anc+13a ; Has10 ; KSG03], ce qui permet d'avoir un mécanisme de réputation passant à l'échelle : lorsque la taille du système augmente, le nombre de tierces parties augmente également, mais pas le nombre d'utilisateurs les constituant ou leur charge de travail. Cependant, pour un mécanisme préservant la vie privée des fournisseurs (propriété 1), il

est impossible d'assigner une tierce partie différente à chaque fournisseur : connaître la tierce partie revient alors à connaître le fournisseur, ce qui permet de les distinguer. Il faut donc une *unique* tierce partie s'occupant de la gestion de *tous* les scores de réputation pour garantir l'indistinguabilité des fournisseurs de service.

Cependant, un seule entité mettant à jour les scores de réputation des fournisseurs de service est un point unique de défaillance : dès que cette entité est compromise ou ne peut s'occuper des mises à jour, les clients ne peuvent plus être tenus au courant de l'évolution des réputations des fournisseurs ; un adversaire pourrait donc attaquer cette entité – empêchant de cette manière les mises à jour des réputations – pour permettre à un fournisseur de bonne réputation de mal se comporter sans voir sa réputation diminuer. Pour empêcher une telle attaque, nous proposons de distribuer cette autorité sur plusieurs utilisateurs du système.

Cette tierce partie n'a pas besoin d'intervenir *pendant* une interaction. Il est suffisant que la mise à jour des réputations ait lieu à intervalles réguliers et non pas en temps réel. Il est cependant nécessaire que l'intervalle de mise à jour ne soit pas trop long, sans quoi un fournisseur peut mal se comporter sans conséquences sur sa réputation pendant longtemps.

Nous appelons *signataires accrédités* les membres de cette tierce partie : ils sont accrédités pour calculer et signer les réputations des fournisseurs de service.

5.3.2 Indéniabilité des témoignages

Garantir l'indéniabilité des témoignages demande de résoudre une contradiction : les clients ne doivent pas savoir qui sont les fournisseurs avec lesquels ils interagissent, mais ils doivent pouvoir témoigner sur le bon fournisseur, même quand celui-ci refuse de donner les informations nécessaires. [3]

Nous proposons une tierce partie dont le rôle est d'assister la construction des témoignages quand le client ou le fournisseur impliqué dans une transaction refuse de collaborer. Similairement à la première tierce partie, nous distribuons celle-ci. De plus, pour garantir que le client ou le fournisseur ne peut obtenir d'informations secrètes en corrompant un seul membre de cette tierce partie, nous utilisons du partage de secret (voir section 2.3) : un seul membre de la tierce partie ne sait rien des informations secrètes, mais un nombre suffisant de ces tierces parties est capable de les reconstruire. Nous appelons donc *porteurs de part* ses membres.

Contrairement à la première tierce partie, celle-ci doit assister le client ou le fournisseur *pendant* les interactions : elle doit donc être en ligne. Cependant, il est possible d'utiliser une tierce partie différente pour chaque interaction.

5.3.3 Instanciation des tierces parties

Nous avons motivé l'utilisation de deux tierces parties distribuées aux rôles distincts. Est-il possible d'utiliser une seule tierce partie distribuée remplissant ces deux rôles ? La tierce partie gérant les scores de réputation doit s'occuper de tous les fournisseurs de service, tandis que celle garantissant l'indéniabilité des témoignages doit être présente lors des interactions. Dans

3. La réciproque est également vraie : les fournisseurs ne doivent pas savoir qui sont leurs clients, mais doivent être capables d'obtenir une preuve de transaction, qui comprend l'invariant, même lorsque le client ne souhaite pas témoigner.

le cas d'un système large-échelle, ces deux fonctionnalités induiraient une charge excessive sur chaque membre de la tierce partie. C'est pourquoi nous utilisons deux tierces parties différentes pour ces deux rôles. La première est unique, et comprend suffisamment d'utilisateurs de confiance pour garantir le calcul de tous les scores de réputation. La deuxième tierce partie s'occupe d'une seule interaction, et permet donc l'émission d'un unique témoignage. Sa sûreté est importante, mais moins cruciale que celle des signataires accrédités : garantir que peu de témoignages peuvent être forgés ou déniés suffit pour obtenir des scores de réputation précis.

Aucun critère particulier ne guide le choix des signataires accrédités. Ils peuvent par exemple être choisis par l'autorité centrale qui contrôle l'enregistrement des utilisateurs. Les fournisseurs les plus actifs du système sont un choix naturel pour être signataires accrédités : ils ont intérêt à ce que le système perdure afin de continuer à effectuer des transactions. Nous notons n_{SA} le nombre de signataires accrédités.

Les signataires accrédités doivent être disponibles à intervalles réguliers pour mettre à jour les scores de réputation ; dans la suite, nous appelons *round* ces intervalles. Lorsqu'un nouveau round commence, chaque signataire accrédité agrège les témoignages émis au round précédent, calcule les nouveaux scores de réputation des fournisseurs de service et les signe. [4] Un compromis raisonnable est de signer les réputations tous les jours : les signataires accrédités ne sont pas démesurément sollicités, et les mises à jour des réputations sont suffisamment réactives. Un fournisseur utilise les signatures émises par les signataires accrédités pour prouver son score de réputation ; une majorité de signatures $t_{\mathrm{SA}} = \lceil n_{\mathrm{SA}}/2 \rceil$ suffit, ce qui empêche toute collusion minoritaire de forger des scores de réputation. Dans la suite, nous supposons qu'une majorité de signataires accrédités sont honnêtes.

Comme expliqué précédemment, les porteurs de part sont choisis pour une interaction spécifique. Nous proposons de les choisir parmi tous les fournisseurs de service du système – pas nécessairement parmi les plus actifs. Le client et le fournisseur d'une interaction choisissent conjointement une graine, qui sert ensuite à choisir les porteurs de part de manière pseudo-aléatoire. Plus précisément, notons N le nombre total de fournisseurs de service, parmi lesquels sont choisis les porteurs de part, et n_{PP} le nombre de porteurs de part désirés. La fonction ChooseSC renvoie un ensemble de n_{PP} porteurs de part choisis pseudo-aléatoirement ; cette fonction est définie par

$$\mathrm{ChooseSC}(\mathrm{seed}, N, n_{\mathrm{PP}}) = \left\{ \left\lfloor H(\mathrm{seed} \| i) \times \frac{N}{2^h} \right\rfloor, i \in \{0, \ldots, n' - 1\} \right\},$$

où H est une fonction de hachage, par exemple SHA-256 [Nat12], dont la sortie est une chaîne de h bits, et n' est choisi de manière à ce que l'ensemble résultant contienne exactement n_{PP} porteurs de part différents. Cette fonction divise l'ensemble $\{0, 1, \ldots, 2^h - 1\}$ en N intervalles de tailles équivalentes, et choisit une séquence pseudo-aléatoire d'éléments de $\{0, 1, \ldots, 2^h - 1\}$ à partir de la graine ; chaque élément de cette séquence renvoie vers un intervalle. Les n_{PP} premiers différents intervalles sont choisis, ce qui détermine les porteurs de part utilisés. Afin que ceux-ci ne soient déterminés ni par le client, ni par le fournisseur, il est nécessaire que la graine

4. Dans ce chapitre, nous ne faisons aucune hypothèse sur le moteur de réputation employé. Par exemple, le moteur proposé au chapitre précédent peut être utilisé.

seed soit générée à partir de nonces choisis *indépendamment* par le client et le fournisseur. Nous précisons la génération de la graine lors de la description du protocole d'interaction, en section 5.4.1.

Comme les porteurs de part sont moins critiques que les signataires accrédités et que les gestionnaires de score des mécanismes précédents, il n'est pas nécessaire d'avoir une probabilité de collusion aussi faible que celle utilisée pour le choix des gestionnaires de score (voir section 4.2). Nous considérons que $p_{\max} = 2^{-20}$ suffit : cela correspond à une interaction sur un million où un témoignage peut soit être émis avant la transaction, soit être dénié ; les scores de réputation des fournisseurs seront toujours proches de leur comportement. Avec ce paramètre, il faut choisir $n_{\mathsf{PP}} = 28$ porteurs de part quand $f : n \mapsto \lceil n/3 \rceil - 1$, et $n_{\mathsf{PP}} = 15$ quand $f : n \mapsto \lceil n/2 \rceil - 1$.

5.4 Mécanismes de réputation préservant la vie privée des clients et des fournisseurs de service

Nous présentons maintenant deux mécanismes de réputation garantissant toutes les propriétés de vie privée et de sécurité énoncées au chapitre 1. Dans un premier temps, nous décrivons la mise en place des utilisateurs, qui diffère légèrement de celle des deux mécanismes précédents. Nous présentons ensuite les deux mécanismes, classique et optimiste, qui préservent également la vie privée du fournisseur ; ces deux mécanismes ne diffèrent que par le schéma de partage de secret – classique ou optimiste – employé. La différence entre ces deux mécanismes et les deux mécanismes ne préservant pas la vie privée des fournisseurs de service concerne tous les éléments envoyés par le fournisseur ; comme celui-ci ne doit pas être identifiable par le client avant la transaction, tous les éléments l'identifiant doivent être masqués par des preuves NIZK. Par exemple, l'invariant ne peut pas être calculé tel quel par le client, c'est pourquoi nous utilisons le pré-invariant et l'invariant masqué introduits en section 5.1.1.

Ces deux mécanismes et leurs versions préliminaires ont donné lieu à plusieurs publications [Anc+13b ; Anc+14 ; Laj+15a ; Laj+15b].

5.4.1 Mise en place des utilisateurs

Tout comme précédemment, nous considérons une autorité centrale C possédant une paire de clés de signature automorphe $(\mathsf{sk}_C, \mathsf{vk}_C)$. En plus de cette autorité, nous considérons quatre types d'utilisateurs : des fournisseurs de service, des clients, des signataires accrédités ainsi que des porteurs de part. La table 5.3 présente les éléments possédés par chaque utilisateur.

Un fournisseur possède une paire de clés de signature automorphe $(\mathsf{sk}_{\mathsf{FS}}, \mathsf{vk}_{\mathsf{FS}})$, un identifiant $\mathsf{Id}_{\mathsf{FS}} \in \mathbb{G}_1$ qui lui est attribué par C, ainsi qu'un certificat $\mathsf{cert}_{\mathsf{FS}}$ émis par C portant sur $\mathsf{vk}_{\mathsf{FS}}$ et $\mathsf{Id}_{\mathsf{FS}}$.

La mise en place des clients est inchangée : ils possèdent une paire de clés de signature automorphe $(\mathsf{sk}_{\mathsf{Cl}}, \mathsf{vk}_{\mathsf{Cl}})$, un identifiant $\mathsf{id}_{\mathsf{Cl}} \in \mathbb{Z}_p$ qui leur est attribué par C, ainsi qu'un certificat $\mathsf{cert}_{\mathsf{Cl}}$ émis par C portant sur $\mathsf{vk}_{\mathsf{Cl}}$ et $\mathsf{id}_{\mathsf{Cl}}$.

Les signataires accrédités sont au nombre de n_{SA}, et chacun possède une paire de clés de signature automorphe $(\mathsf{sk}_{\mathsf{SA}_i}, \mathsf{vk}_{\mathsf{SA}_i})$ ainsi qu'un certificat portant sur $\mathsf{vk}_{\mathsf{SA}_i}$. Ils utilisent cette

	C	SA	PP	Client		Fournisseur	
				Élément	Engagement	Élément	Engagement
Clé de sig.	$\mathbf{sk}_C$	$\mathbf{sk}_{\mathsf{SA}}$	$\mathbf{sk}_{\mathsf{PP}}$	$\mathbf{sk}_{\mathsf{Cl}}$		$\mathbf{sk}_{\mathsf{FS}}$	
Clé de vérif.	vk_C	$\mathsf{vk}_{\mathsf{SA}}$	$\mathsf{vk}_{\mathsf{PP}}$	$\mathsf{vk}_{\mathsf{Cl}}$	$\mathsf{nym}_{\mathsf{Cl}}$	$\mathsf{vk}_{\mathsf{FS}}$	$\mathsf{nym}_{\mathsf{FS}}$
Clé de chiff.			$\mathsf{ek}_{\mathsf{PP}}$				
Clé de déchiff.			$\mathbf{dk}_{\mathsf{PP}}$				
Identifiant				$\mathsf{id}_{\mathsf{Cl}}$	$C_{\mathsf{id}_{\mathsf{Cl}}}$	$\mathsf{Id}_{\mathsf{FS}}$	$C_{\mathsf{Id}_{\mathsf{FS}}}$
Certificat		$\mathsf{cert}_{\mathsf{SA}}$	$\mathsf{cert}_{\mathsf{PP}}$	$\mathsf{cert}_{\mathsf{Cl}}$	$C_{\mathsf{cert}_{\mathsf{Cl}}}$	$\mathsf{cert}_{\mathsf{FS}}$	$C_{\mathsf{cert}_{\mathsf{FS}}}$
(vérifié avec)		$(\mathsf{vk}_{\mathsf{SA}})$	$(\mathsf{vk}_{\mathsf{PP}}, \mathsf{ek}_{\mathsf{PP}})$	$(\mathsf{vk}_{\mathsf{Cl}}, \mathsf{id}_{\mathsf{Cl}})$	$(\pi_{\mathsf{cert}_{\mathsf{Cl}}}, \mathsf{nym}_{\mathsf{Cl}}, C_{\mathsf{id}_{\mathsf{Cl}}})$	$(\mathsf{vk}_{\mathsf{FS}}, \mathsf{Id}_{\mathsf{FS}})$	$(\pi_{\mathsf{cert}_{\mathsf{FS}}}, \mathsf{nym}_{\mathsf{FS}}, C_{\mathsf{Id}_{\mathsf{FS}}})$
Réputation						$\mathsf{rep}_{\mathsf{FS}}$	
Signature						$(\sigma_{\mathsf{rep},i})_i$	$(C_{\sigma_{\mathsf{rep},i}})_i$
(vérifiées avec)						$(\mathsf{vk}_{\mathsf{FS}}, \mathsf{rep}_{\mathsf{FS}})$	$(\mathsf{nym}_{\mathsf{FS}}, \mathsf{rep}_{\mathsf{FS}}, \pi_{\mathsf{rep}_{\mathsf{FS}}})$

TABLE 5.3 – Éléments des utilisateurs – les éléments en gras sont secrets

paire de clés pour signer les réputations des fournisseurs. Par exemple, si $\mathsf{rep}_{\mathsf{FS}}$ est la réputation d'un fournisseur, la signature du i-ième signataire accrédité au round rnd est définie par

$$\sigma_{\mathsf{rep},i} = \mathrm{SPSign}\left(\mathsf{sk}_{\mathsf{SA}_i}, \langle \mathsf{rep}_{\mathsf{FS}}, \mathsf{Id}_{\mathsf{FS}}, \mathsf{rnd}\rangle\right).$$

Les fournisseurs utilisent une majorité $t_{\mathsf{SA}} = \lceil n_{\mathsf{SA}}/2 \rceil$ de ces signatures pour construire leurs preuves de réputation $\pi_{\mathsf{rep}_{\mathsf{FS}}}$.

Les porteurs de part sont similaires aux gestionnaires de score des précédents mécanismes : ils possèdent une paire de clés de signature classique $(\mathsf{sk}_{\mathsf{PP}_i}, \mathsf{vk}_{\mathsf{PP}_i})$ et une paire de clés de chiffrement classique $(\mathsf{dk}_{\mathsf{PP}_i}, \mathsf{ek}_{\mathsf{PP}_i})$. L'autorité centrale leur fournit également un certificat $\mathsf{cert}_{\mathsf{PP}_i}$ portant sur $\mathsf{vk}_{\mathsf{PP}_i}$ et $\mathsf{ek}_{\mathsf{PP}_i}$. Lors d'une interaction, le client et le fournisseur choisiront n_{PP} porteurs de part : pour le schéma de partage classique, il faut $n_{\mathsf{PP}} = 28$, tandis que $n_{\mathsf{PP}} = 15$ suffit au cas optimiste (voir section 5.3.3).

Pour générer leurs pseudonymes, les clients et les fournisseurs procèdent comme les clients des deux précédents mécanismes, c'est-à-dire en s'engageant sur leurs identifiants :

$$\begin{aligned}
\mathsf{nym}_{\mathsf{Cl}} &= \mathrm{Com}(\mathsf{vk}_{\mathsf{Cl}}, _) \\
\mathsf{nym}_{\mathsf{FS}} &= \mathrm{Com}\left(\mathsf{vk}_{\mathsf{FS}}, (r_{\mathsf{FS}}, r'_{\mathsf{FS}})\right) \\
C_{\mathsf{id}_{\mathsf{Cl}}} &= \mathrm{Com}(\mathsf{id}_{\mathsf{Cl}}, _) \\
C_{\mathsf{Id}_{\mathsf{FS}}} &= \mathrm{Com}\left(\mathsf{Id}_{\mathsf{FS}}, (r_{\mathsf{Id}_{\mathsf{FS}}}, r'_{\mathsf{Id}_{\mathsf{FS}}})\right) \\
C_{\mathsf{cert}_{\mathsf{Cl}}} &= \mathrm{Com}(\mathsf{cert}_{\mathsf{Cl}}, _) \\
C_{\mathsf{cert}_{\mathsf{FS}}} &= \mathrm{Com}(\mathsf{cert}_{\mathsf{FS}}, _),
\end{aligned}$$

où $r_{\mathsf{FS}}, r'_{\mathsf{FS}}, r_{\mathsf{Id}_{\mathsf{FS}}}, r'_{\mathsf{Id}_{\mathsf{FS}}} \xleftarrow{\mathrm{R}} \mathbb{Z}_p$, et construisent des preuves $\pi_{\mathsf{cert}_{\mathsf{Cl}}}$ et $\pi_{\mathsf{cert}_{\mathsf{FS}}}$ de validité de leur certificat, comme décrit en section 5.2.1. Comme précédemment, ces engagements et preuves sont utilisés pour une seule interaction.

Finalement, les fournisseurs calculent un pré-invariant en choisissant un aléa $r_{\mathsf{pre\_inv}} \xleftarrow{\mathrm{R}} \mathbb{Z}_p$,

et construisent une preuve $\pi_{\text{pre\_inv}}$ de son calcul :

$$\begin{aligned}\text{pre\_inv} &= \text{Pre\_inv}(\text{Id}_{\text{FS}}, r_{\text{pre\_inv}})\\ \pi_{\text{pre\_inv}} &= \text{NIZK}\left\{\text{Id}_{\text{FS}}, r_{\text{pre\_inv}} : \text{pre\_inv} = \text{Pre\_inv}(\text{Id}_{\text{FS}}, r_{\text{pre\_inv}})\right\}.\end{aligned}$$

Ils calculent également des engagements sur les signatures reçues des signataires accrédités, et une preuve de réputation $\pi_{\text{rep}_{\text{FS}}}$ qui garantit la validité de t_{SA} signatures, sans révéler ni les signatures, ni l'identifiant du fournisseur :

$$\begin{aligned}\forall 1 \leqslant i \leqslant n_{\text{SA}}, \quad C_{\sigma_{\text{rep},i}} &= \text{Com}(\sigma_{\text{rep},i}, \_)\\ \pi_{\text{rep}_{\text{FS}}} &= \text{NIZK}\left\{\text{Id}_{\text{FS}}, \{\sigma_{\text{rep},i_j}\}_{1\leqslant j\leqslant t_{\text{SA}}} :\right.\\ &\qquad \left.\wedge_{1\leqslant j\leqslant t_{\text{SA}}} \text{SPVerif}(\sigma_{\text{rep},i_j}, \langle \text{Id}_{\text{FS}}, \text{rep}_{\text{FS}}, \text{rnd}\rangle, \text{vk}_{\text{SA}_{i_j}})\right\}.\end{aligned}$$

Ces éléments permettent aux fournisseurs de prouver leur réputation tout en préservant leur vie privée.

5.4.2 Mécanisme préservant la vie privée des utilisateurs

Comme nous l'avons brièvement évoqué au début de cette section, une interaction entre un client et un fournisseur se déroule en quatre étapes : dans un premier temps, le fournisseur prouve sa réputation au client, et le client s'authentifie auprès du fournisseur ; les porteurs de part sont également choisis pendant cette étape. Une fois les porteurs de part choisis, le client et le fournisseur partagent leurs secrets, ce qui garantit les propriétés d'indéniabilité. La transaction peut alors se dérouler. Après cela, le client et le fournisseur peuvent construire le témoignage ; cette étape peut se dérouler de trois manières différentes suivant les comportements du fournisseur et du client. Finalement, et de manière désynchronisée des interactions, les signataires accrédités mettent les scores de réputation des fournisseurs à jour et signent leurs réputations.

Dans cette section, nous utilisons le partage de secret vérifiable classique, c'est-à-dire que nous n'essayons pas de construire un mécanisme optimiste. Nous verrons ensuite comment adapter ce mécanisme pour en obtenir un optimiste.

Preuve de réputation

Lorsqu'un client est intéressé par le service proposé par un fournisseur, il commence par s'authentifier en lui envoyant nym_{Cl}, $C_{\text{id}_{\text{Cl}}}$, $C_{\text{cert}_{\text{Cl}}}$ et $\pi_{\text{cert}_{\text{Cl}}}$. Si la preuve est valide, le fournisseur choisit un nonce $s_{\text{PP}} \xleftarrow{\text{R}} \mathbb{Z}_p$ sur lequel il s'engage en calculant $C_{\text{PP}} = H(\texttt{00}\|s_{\text{PP}})$, et répond avec nym_{FS}, $C_{\text{Id}_{\text{FS}}}$, $C_{\text{cert}_{\text{FS}}}$, $\pi_{\text{cert}_{\text{FS}}}$, rep_{FS}, $(C_{\sigma_{\text{rep},i_j}})_{1\leqslant j<t_{\text{SA}}}$, π_{rep}, pre_inv, $\pi_{\text{pre\_inv}}$ et C_{PP}. Le client vérifie les trois preuves et retient C_{PP}. Il choisit également un nonce $r_{\text{PP}} \xleftarrow{\text{R}} \mathbb{Z}_p$, calcule l'invariant masqué à l'aide d'un aléa $r_{\text{masked\_inv}} \xleftarrow{\text{R}} \mathbb{Z}_p$, et prouve ce calcul en masquant à la fois l'invariant masqué et l'aléa utilisé, pour lesquels il s'engage :

$$\begin{aligned}\text{masked\_inv} &= \text{Mask}(\text{pre\_inv}, \text{id}_{\text{Cl}}, r_{\text{masked\_inv}})\\ \pi_{C_{\text{masked\_inv}}} &= \text{NIZK}\left\{\text{id}_{\text{Cl}}, \text{masked\_inv}, r_{\text{masked\_inv}} :\right.\\ &\qquad \left.\text{masked\_inv} = \text{Mask}(\text{pre\_inv}, \text{id}_{\text{Cl}}, r_{\text{masked\_inv}})\right\}.\end{aligned}$$

Finalement, il construit une signature proxy anonyme ς_{Cl} définie par

$$\varsigma_{\mathsf{Cl}} = \mathrm{APSign}\left(\mathsf{sk}_{\mathsf{Cl}}, H(C_{\mathsf{PP}}, r_{\mathsf{PP}}, \mathsf{nym}_{\mathsf{FS}}, \mathsf{pre\_inv}, C_{\mathsf{masked\_inv}})\right),$$

et envoie le nonce, les engagements sur l'invariant masqué et son aléa, la preuve de ce calcul, ainsi que la signature au fournisseur. Celui-ci vérifie la preuve et la signature, et construit ς_{FS} définie par

$$\varsigma_{\mathsf{FS}} = \mathrm{APSign}\left(\mathsf{sk}_{\mathsf{FS}}, H(s_{\mathsf{PP}}, r_{\mathsf{PP}}, \mathsf{nym}_{\mathsf{Cl}}, \mathsf{pre\_inv}, C_{\mathsf{masked\_inv}})\right).$$

Il révèle s_{PP} au client, et lui envoie également ς_{FS}. Le client vérifie que $C_{\mathsf{PP}} = H(\mathtt{00} \| s_{\mathsf{PP}})$. L'engagement du fournisseur sur s_{PP} garantit que le fournisseur et le client ont choisi leurs nonces indépendamment : le client a choisi r_{PP} sans connaître s_{PP}, et réciproquement. Les deux signatures portent sur des éléments « frais », c'est-à-dire qu'il n'est pas possible de rejouer une ancienne signature ; ces deux signatures garantissent donc que le client désire interagir avec le fournisseur, et vice-versa. La figure 5.7 décrit cet échange.

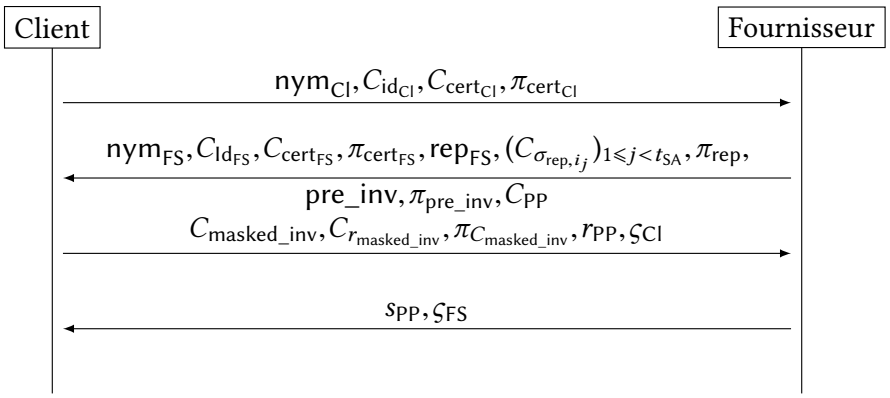


FIGURE 5.7 – Preuve de réputation

Une fois que les nonces ont été échangés, les porteurs de part peuvent être choisis en utilisant la fonction ChooseSC définie en section 5.3.3. Cette fonction est appliquée à la graine

$$H(\mathtt{01} \| s_{\mathsf{PP}} \| r_{\mathsf{PP}} \| \mathsf{nym}_{\mathsf{Cl}} \| \mathsf{nym}_{\mathsf{FS}}),$$

ce qui détermine les n_{PP} porteurs de part. Cette graine sert également d'identifiant de transaction ; à cet effet, nous la notons $\mathsf{id}_{\mathsf{trans}}$.

Remarque (Attaques par extension de longueur)
Le `00` et le `01` de C_{PP} et $\mathsf{id}_{\mathsf{trans}}$ empêchent les attaques par extension de longueur [DR09], qui permettraient sinon au client de choisir r_{PP} de sorte que les $(H(s_{\mathsf{PP}} \| r_{\mathsf{PP}} \| \mathsf{nym}_{\mathsf{Cl}} \| \mathsf{nym}_{\mathsf{FS}} \| i))_{i \in \mathbb{N}}$ donnent un ensemble de porteurs de part lui convenant. ■

Partage des secrets

Le partage des secrets se décompose en deux sous-protocoles exécutés simultanément : le fournisseur partage son secret, – son identifiant – pendant que le client partage le sien – l'invariant masqué. Une fois les deux partages vérifiés, la transaction peut avoir lieu.

Partage de l'identifiant du fournisseur Pour partager son identifiant, le fournisseur choisit un polynôme aléatoirement et construit des preuves comme décrit en section 5.1.2 ; le polynôme est $Q : z \mapsto \mathsf{Id}_{\mathsf{FS}} \cdot \prod_{j=1}^{t_{\mathsf{PP}}-1} A_j^{(z^j)}$, où les A_j sont choisis aléatoirement, et les parts sont les $S_i = Q(i)$ pour $1 \leqslant i \leqslant n_{\mathsf{PP}}$. Les preuves sont les π_{S_i}, et les engagements sur les A_j sont les C_{A_j} ; l'engagement sur le secret est $C_{\mathsf{Id}_{\mathsf{FS}}}$, et a été calculé précédemment. Le fournisseur chiffre les parts pour les porteurs de part :

$$E_{S_i} = \mathrm{Enc}_{\mathsf{ek}_{\mathsf{PP}_i}}(S_i).$$

Finalement, le fournisseur envoie $\mathsf{id}_{\mathsf{trans}}$, i, E_{S_i}, $C_{\mathsf{Id}_{\mathsf{FS}}}$, $(C_{A_j})_j$ et π_{S_i} à chaque porteur de part. Quand un porteur de part reçoit une part valide, il envoie une confirmation $\mathsf{id}_{\mathsf{trans}}$, $\sigma_{\mathsf{PP}_i,\mathsf{CI}}$ au client, où la signature est définie par

$$\sigma_{\mathsf{PP}_i,\mathsf{CI}} = \mathrm{Sign}\left(\mathsf{sk}_{\mathsf{PP}_i}, H\Big(i, C_{\mathsf{Id}_{\mathsf{FS}}}, (C_{A_j})_j\Big)\right).$$

Le fournisseur a déjà prouvé la validité du secret au client grâce à la preuve $\pi_{\mathsf{cert}_{\mathsf{FS}}}$; il suffit donc qu'il lui envoie les $(C_{A_j})_j$. Finalement, le client accepte le partage dès qu'il a reçu $2\lceil n_{\mathsf{PP}}/3 \rceil - 1$ confirmations valides. La figure 5.8 décrit le partage du fournisseur.

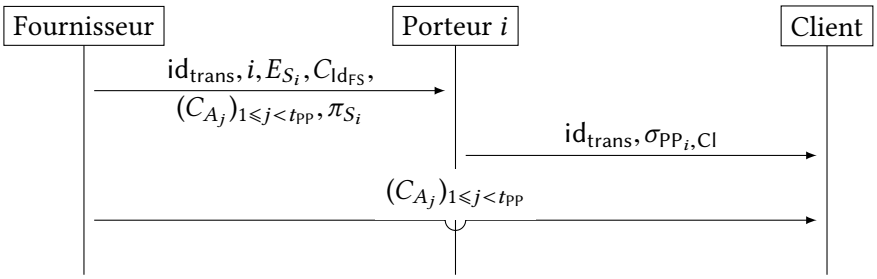


FIGURE 5.8 – Partage de l'identifiant du fournisseur

Partage de l'invariant masqué Le partage de l'invariant masqué est légèrement différent. En effet, celui-ci est un élément de $\mathbb{G}_1 \times \mathbb{G}_1$; pour le partager, il faut donc partager ses deux composants, $\mathsf{masked\_inv}_{\mathbf{1}}$ et $\mathsf{masked\_inv}_{\mathbf{2}}$. Le client choisit à cet effet deux polynômes, $R_{\mathbf{1}} : z \mapsto \mathsf{masked\_inv}_{\mathbf{1}} \cdot \prod_{j=1}^{t_{\mathsf{PP}}-1} B_{j_{\mathbf{1}}}{}^{(i^j)}$ et $R_{\mathbf{2}} : z \mapsto \mathsf{masked\_inv}_{\mathbf{2}} \cdot \prod_{j=1}^{t_{\mathsf{PP}}-1} B_{j_{\mathbf{2}}}{}^{(i^j)}$. Les parts sont les $T_i = \Big(R_{\mathbf{1}}(i), R_{\mathbf{2}}(i)\Big)$ pour $1 \leqslant i \leqslant n_{\mathsf{PP}}$, et leurs preuves sont également doublées :

$$\begin{aligned}\pi_{T_i} = \mathrm{NIZK}\Big\{&\mathsf{masked\_inv}, (B_{1_{\mathbf{k}}}, \ldots, B_{(t_{\mathsf{PP}}-1)_{\mathbf{k}}})_{\mathbf{k} \in \{\mathbf{1},\mathbf{2}\}} : \\ &T_{i\,\mathbf{1}} = \mathsf{masked\_inv}_{\mathbf{1}} \cdot \prod_{j=1}^{t-1} B_{j_{\mathbf{1}}}{}^{(i^j)} \\ \wedge\, &T_{i\,\mathbf{2}} = \mathsf{masked\_inv}_{\mathbf{2}} \cdot \prod_{j=1}^{t-1} B_{j_{\mathbf{2}}}{}^{(i^j)}\Big\}.\end{aligned}$$

Finalement, le client chiffre les parts pour les porteurs de part : $E_{T_i} = \mathrm{Enc}_{\mathsf{ek}_{\mathsf{PP}_i}}(T_i), 1 \leqslant i \leqslant n_{\mathsf{PP}}$. Le client envoie donc $\mathsf{id}_{\mathsf{trans}}, i, E_{T_i}, C_{\mathsf{masked\_inv}}, (C_{B_j})_{1 \leqslant j < t_{\mathsf{PP}}}$ et π_{T_i} à chaque porteur de part. [5] Si la preuve reçue par un porteur de part est valide, il répond en signant $H(i, C_{\mathsf{masked\_inv}}, (C_{B_j})_j)$:

$$\sigma_{\mathsf{PP}_i,\mathsf{FS}} = \mathrm{Sign}\left(\mathsf{sk}_{\mathsf{PP}_i}, H(i, C_{\mathsf{masked\_inv}}, (C_{B_j})_{1 \leqslant j < t_{\mathsf{PP}}})\right).$$

En parallèle, le client envoie $(C_{B_j})_j$ au fournisseur. Celui-ci accepte le partage si la preuve est valide et une fois qu'il a reçu $2\lceil n_{\mathsf{PP}}/3 \rceil - 1$ confirmations valides. La figure 5.9 décrit cet échange.

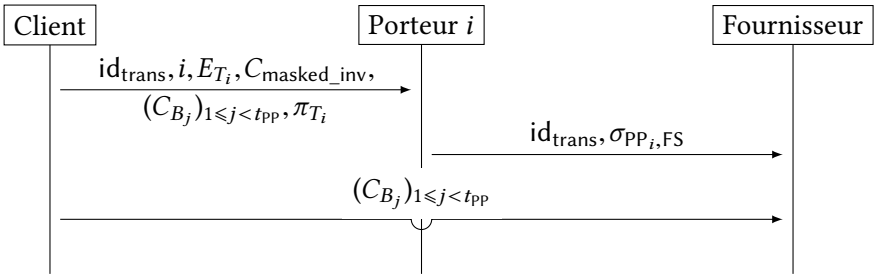


FIGURE 5.9 – Partage de l'invariant masqué

Une fois que les deux partages ont été acceptés, la transaction peut avoir lieu.

Construction du témoignage

Après la transaction, le client a l'occasion de témoigner. S'il le désire, il procède comme décrit au scénario A ; sinon, le fournisseur peut obtenir la preuve de transaction en lançant le scénario B. Finalement, si le fournisseur n'aide pas le client à émettre son témoignage, le scénario C est déclenché.

Scénario A – cas nominal Pour émettre un témoignage, le client commence par choisir une note ρ et la signer avec une signature proxy anonyme ς_ρ :

$$\varsigma_\rho = \mathrm{APSign}\left(\mathsf{sk}_{\mathsf{Cl}}, H(\mathsf{id}_{\mathsf{trans}}, \rho, \mathsf{masked\_inv})\right).$$

Le client prouve également le calcul de masked_inv, sans le masquer :

$$\pi_{\mathsf{masked\_inv}} = \mathrm{NIZK}\Big\{\mathsf{id}_{\mathsf{Cl}}, r_{\mathsf{masked\_inv}} : \mathsf{masked\_inv} = \mathrm{Mask}(\mathsf{pre\_inv}, \mathsf{id}_{\mathsf{Cl}}, r_{\mathsf{masked\_inv}})\Big\}.$$

5. Rappelons que $C_{B_j} = (C_{B_{j_1}}, C_{B_{j_2}})$ (voir la description des notations en section 2.1.1).

Remarque
L'aléa $r_{\text{masked\_inv}}$ reste masqué. En effet, s'il est révélé, le fournisseur peut calculer

$$\text{masked\_inv}_{\mathbf{1}} \cdot {G_1}^{-r_{\text{masked\_inv}}} = {Y_1}^{\text{id}_{\text{Cl}}}.$$

Deux fournisseurs différents pourraient alors comparer leurs clients, ce qui va contre le respect de leur vie privée (propriété 2). ■

Le client envoie alors au fournisseur un message m_1 contenant tous les éléments nécessaires au calcul de l'invariant, c'est-à-dire $m_1 = (\text{id}_{\text{trans}}, \rho, \text{masked\_inv}, \varsigma_\rho, \pi_{\text{masked\_inv}})$. Si la signature et la preuve sont valides, le fournisseur calcule l'invariant :

$$\text{inv} = \text{Unmask}(\text{masked\_inv}, r_{\text{pre\_inv}}).$$

Le fournisseur répond au client un message m_2 dans lequel il s'identifie, et dans lequel il signe la note du client : $m_2 = (\text{Id}_{\text{FS}}, \text{vk}_{\text{FS}}, \text{cert}_{\text{FS}}, \text{inv}, \sigma_{\rho,\text{FS}}, \pi_{\text{FS}})$, où la signature est définie par

$$\sigma_{\rho,\text{FS}} = \text{SPSign}\left(\text{sk}_{\text{FS}}, H(\text{id}_{\text{trans}}, \varsigma_\rho, \text{inv})\right),$$

et la preuve π_{FS} contient les aléas utilisés par le fournisseur pour s'engager sur ses identifiants et sur le pré-invariant, c'est-à-dire $\pi_{\text{FS}} = (r_{\text{FS}}, r'_{\text{FS}}, r_{\text{Id}_{\text{FS}}}, r'_{\text{Id}_{\text{FS}}}, r_{\text{pre\_inv}})$. Puisque le fournisseur n'est plus anonyme, sa signature est automorphe et non proxy anonyme.

À partir de tous ces éléments, aussi bien le client que le fournisseur peuvent émettre le témoignage. Les éléments composant le témoignage sont présentés en première colonne de la table 5.4. La figure 5.10 présente l'échange entre le client et le fournisseur.

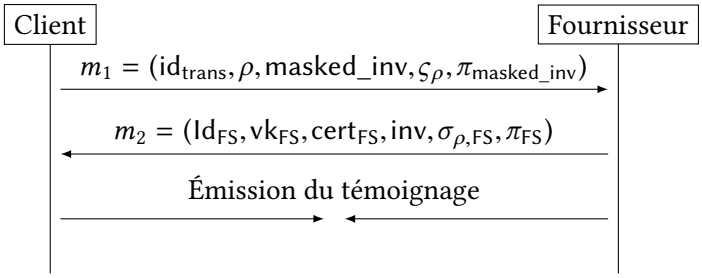


FIGURE 5.10 – Construction et émission du témoignage dans le scénario A

Si jamais le fournisseur ne reçoit pas le message m_1 et désire obtenir une preuve de transaction, il contacte les porteurs de part et déclenche le scénario B. Si le client a envoyé le message m_1 mais ne reçoit pas le message m_2, il lance le scénario C pour être capable d'émettre un témoignage.

Scénario B – client malveillant Le fournisseur demande les parts des porteurs de part. Ceux-ci laissent d'abord le temps au client d'envoyer le message m_1 pour qu'il puisse noter le fournisseur. Si le client ne répond toujours pas, les porteurs de part envoient leur part ainsi que sa preuve de validité au fournisseur. Comme nous supposons qu'au plus $\lceil n_{\text{PP}}/3 \rceil - 1$ porteurs

	Scénario A	Scénario B	Scénario C
FS	$\mathsf{Id}_{\mathsf{FS}}$, $\mathsf{vk}_{\mathsf{FS}}$, $\mathsf{nym}_{\mathsf{FS}}$, $C_{\mathsf{Id}_{\mathsf{FS}}}$, $\mathsf{cert}_{\mathsf{FS}}$, π_{FS}	$\mathsf{Id}_{\mathsf{FS}}$, $\mathsf{vk}_{\mathsf{FS}}$, $\mathsf{nym}_{\mathsf{FS}}$, $C_{\mathsf{Id}_{\mathsf{FS}}}$, $\mathsf{cert}_{\mathsf{FS}}$, π_{FS}	$\mathsf{Id}_{\mathsf{FS}}$, $\left\{(i_j, S_{i_j}, \pi_{S_{i_j}}), 1 \leqslant j \leqslant t_{\mathsf{PP}}\right\}$, $C_{\mathsf{Id}_{\mathsf{FS}}}$, $\mathsf{nym}_{\mathsf{FS}}$, $(C_{A_j})_{1\leqslant j<t_{\mathsf{PP}}}$, $C_{\mathsf{cert}_{\mathsf{FS}}}$, $\pi_{\mathsf{cert}_{\mathsf{FS}}}$
Client	$\mathsf{nym}_{\mathsf{Cl}}$, $C_{\mathsf{id}_{\mathsf{Cl}}}$, $C_{\mathsf{cert}_{\mathsf{Cl}}}$, $\pi_{\mathsf{cert}_{\mathsf{Cl}}}$	$\mathsf{nym}_{\mathsf{Cl}}$, $C_{\mathsf{id}_{\mathsf{Cl}}}$, $C_{\mathsf{cert}_{\mathsf{Cl}}}$, $\pi_{\mathsf{cert}_{\mathsf{Cl}}}$	$\mathsf{nym}_{\mathsf{Cl}}$, $C_{\mathsf{id}_{\mathsf{Cl}}}$, $C_{\mathsf{cert}_{\mathsf{Cl}}}$, $\pi_{\mathsf{cert}_{\mathsf{Cl}}}$
Id. trans.	$\mathsf{id}_{\mathsf{trans}}, \varsigma_{\mathsf{FS}}, \varsigma_{\mathsf{Cl}}$	$\mathsf{id}_{\mathsf{trans}}, \varsigma_{\mathsf{FS}}, \varsigma_{\mathsf{Cl}}$	$\mathsf{id}_{\mathsf{trans}}, \varsigma_{\mathsf{FS}}, \varsigma_{\mathsf{Cl}}$
Invariant	pre_inv, masked_inv, inv, $C_{r_{\mathsf{masked\_inv}}}$, $\pi_{\mathsf{masked\_inv}}$	pre_inv, masked_inv, inv, $\left\{(i_j, T_{i_j}, \pi_{T_{i_j}}), 1 \leqslant j \leqslant t_{\mathsf{PP}}\right\}$, $C_{\mathsf{masked\_inv}}$, $(C_{B_j})_{1\leqslant j<t_{\mathsf{PP}}}$, $\pi_{C_{\mathsf{masked\_inv}}}$	inv, Π_{inv}
Note	ρ, ς_ρ, $\sigma_{\rho,\mathsf{FS}}$	N.A.	ρ, ς_ρ, $(\sigma_{\rho,\mathsf{PP}_{i_j}})_{1\leqslant j\leqslant t_{\mathsf{PP}}}$

TABLE 5.4 – Éléments composant le témoignage

de part sont malveillants, au moins $t_{\mathsf{PP}} = \lceil n_{\mathsf{PP}}/3 \rceil$ d'entre eux répondront avec une part valide parmi les $2\lceil n_{\mathsf{PP}}/3 \rceil - 1$ ayant confirmé le partage. À partir des parts, le fournisseur reconstruit l'invariant masqué en interpolant les parts (voir section 2.3.2) :

$$\mathsf{masked\_inv} = \mathrm{Interp}\left(\left\{(i_j, T_{i_j}),\ 1 \leqslant j \leqslant t_{\mathsf{PP}}\right\}\right).$$

Le fournisseur peut alors calculer

$$\mathsf{inv} = \mathrm{Unmask}(\mathsf{masked\_inv}, r_{\mathsf{pre\_inv}})$$

et émettre le témoignage, qui ne contient pas de note du client. La figure 5.11 décrit cette interaction, et les composants du témoignage sont décrits dans la deuxième colonne de la table 5.4.

Scénario C – fournisseur malveillant Si le client ne reçoit pas le message m_2 du fournisseur, il demande les parts des porteurs de part. Cependant, le client doit prouver qu'il a bien envoyé un message m_1 valide au fournisseur. Le client envoie donc $\mathsf{id}_{\mathsf{trans}}$, ρ, ς_ρ. Si la signature est valide, les porteurs de part renvoient leur part au client, c'est-à-dire i, S_i et π_{S_i}, ainsi qu'une signature $\sigma_{\rho,\mathsf{PP}_i}$ sur la note du client :

$$\sigma_{\rho,\mathsf{PP}_i} = \mathrm{Sign}\left(\mathsf{sk}_{\mathsf{PP}_i}, H(\mathsf{id}_{\mathsf{trans}}, \varsigma_\rho, \rho)\right).$$

Le client peut alors reconstruire l'identifiant du fournisseur, calculer

$$\mathsf{inv} = \mathrm{Inv}(\mathsf{Id}_{\mathsf{FS}}, \mathsf{id}_{\mathsf{Cl}}),$$

ainsi qu'une preuve π_{inv} de son calcul :

$$\pi_{\mathsf{inv}} = \textsc{nizk}\left\{\mathsf{id}_{\mathsf{Cl}} : \mathsf{inv} = \mathrm{Inv}(\mathsf{Id}_{\mathsf{FS}}, \mathsf{id}_{\mathsf{Cl}})\right\}.$$

Finalement, le client émet le témoignage, qui comporte les éléments de la troisième colonne de la table 5.4. La figure 5.12 décrit ce scénario.

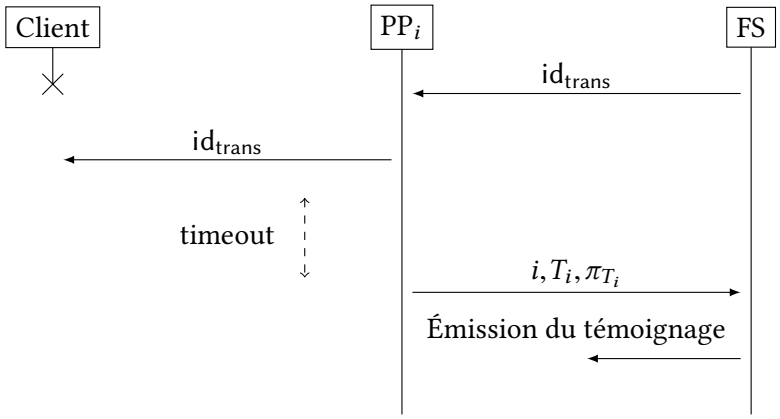


FIGURE 5.11 – Construction et émission du témoignage dans le scénario B

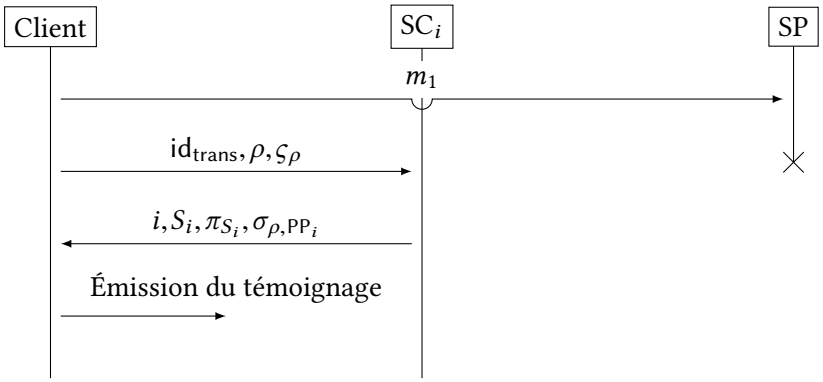


FIGURE 5.12 – Construction et émission du témoignage dans le scénario C

Mise à jour des scores de réputation

À la fin du round rnd, chaque porteur de part agrège les témoignages issus des transactions auxquelles il a participé depuis la fin du round (rnd – 1) et les transmet aux signataires accrédités. Une fois qu'un signataire accrédité a vérifié un témoignage, il conserve uniquement l'identifiant de transaction, l'invariant ainsi que la note du client, c'est-à-dire le triplet (id_{trans}, inv, ρ). À partir de ces témoignages, les signataires accrédités mettent les scores de réputation des fournisseurs de service à jour. Rappelons que ce mécanisme de réputation est indépendant du moteur de réputation ; il est par exemple possible de prendre en compte uniquement les derniers témoignages de chaque client pour limiter les bourrages d'urne, et d'utiliser le moteur de réputation présenté en section 4.4 et de filtrer les témoignages avec la méthode proposée par Whitby et coll. [WJI04].

Une fois que les réputations de tous les fournisseurs ont été mises à jour, les signataires

accrédités les signent :

$$\sigma_{\mathsf{rep},i} = \mathrm{SPSign}\left(\mathsf{sk}_{\mathsf{SA}_i}, \langle \mathsf{Id}_{\mathsf{FS}}, \mathsf{rep}_{\mathsf{FS}}, \mathsf{rnd}\rangle\right),$$

et transmettent ces signatures aux fournisseurs. Ceux-ci peuvent ensuite utiliser ces signatures pour prouver leurs réputation pendant le round (rnd + 1).

5.4.3 Mécanisme optimiste préservant la vie privée des utilisateurs

Le mécanisme précédent nécessite que les porteurs de part soient en ligne lors du partage de secret. Nous en présentons maintenant une variante, remplaçant le partage de secret vérifiable (section 5.1.2) par du partage de secret vérifiable optimiste (section 5.1.3). Les seules étapes modifiées sont le partage des secrets ainsi que l'émission du témoignage dans les scénarios B et C.

Partage des secrets

Partage de l'identifiant du fournisseur Similairement au mécanisme précédent, le fournisseur choisit un polynôme $Q : z \mapsto \mathsf{Id}_{\mathsf{FS}} \cdot \prod_{j=1}^{t_{\mathsf{PP}}-1} A_j{}^{(z^j)}$ aléatoirement. Les parts sont les $S_i = Q(i)$ pour $1 \leqslant i \leqslant n_{\mathsf{PP}}$. Le fournisseur s'engage sur les parts, les chiffre et les signe pour les porteurs de part :

$$\begin{aligned} C_{S_i} &= \mathrm{Com}(S_i, \_) \\ E_{S_i} &= \mathrm{Enc}_{\mathsf{ek}_i}(S_i) \\ \sigma_{S_i} &= \mathrm{SPSign}(\mathsf{sk}_{\mathsf{FS}}, S_i) \\ C_{\sigma_{S_i}} &= \mathrm{Com}(\sigma_{S_i}, \_), \end{aligned}$$

pour $1 \leqslant i \leqslant n_{\mathsf{PP}}$. Finalement, le fournisseur prouve le partage en calculant

$$\begin{aligned} \pi_{E_{S_i}} = \textsc{nizk}\Big\{ & S_i, \sigma_{S_i}, \mathsf{Id}_{\mathsf{FS}}, A_1, \ldots, A_{t_{\mathsf{PP}}-1} : \\ & E_{S_i} = \mathrm{Enc}_{\mathsf{ek}_i}(S_i) \\ & \wedge S_i = \mathsf{Id}_{\mathsf{FS}} \cdot \prod_{j=1}^{t_{\mathsf{PP}}-1} A_j{}^{(i^j)} \\ & \wedge \mathrm{SPVerif}(\sigma_{S_i}, S_i, \mathsf{vk}_{\mathsf{FS}}) \Big\}, \end{aligned}$$

pour $1 \leqslant i \leqslant n_{\mathsf{PP}}$. Le fournisseur envoie finalement $C_{A_1}, \ldots, C_{A_{t_{\mathsf{PP}}-1}}$, $(E_{S_i}, C_{S_i}, C_{\sigma_{S_i}}, \pi_{E_{S_i}})_i$ au client. Le client ayant déjà vérifié la validité de $C_{\mathsf{Id}_{\mathsf{FS}}}$ en même temps que la preuve de réputation du fournisseur, il accepte le partage si les n_{PP} preuves sont valides.

Partage de l'invariant masqué Comme précédemment, le client choisit deux polynômes $R_{\mathbf{1}} : z \mapsto \mathsf{masked\_inv}_{\mathbf{1}} \cdot \prod_{j=1}^{t_{\mathsf{PP}}-1} B_{j_{\mathbf{1}}}{}^{(z^j)}$ et $R_{\mathbf{2}} : z \mapsto \mathsf{masked\_inv}_{\mathbf{2}} \cdot \prod_{j=1}^{t_{\mathsf{PP}}-1} B_{j_{\mathbf{2}}}{}^{(z^j)}$ aux coefficients aléatoires. Les parts sont les $T_i = \left(R_{\mathbf{1}}(i), R_{\mathbf{2}}(i)\right)$ pour $1 \leqslant i \leqslant n_{\mathsf{PP}}$. Le client s'engage sur les

parts, les chiffre, les signe et prouve leur validité pour les porteurs de part :

$$\begin{aligned}
C_{T_i} &= \mathrm{Com}(T_i, _) \\
E_{T_i} &= \mathrm{Enc}_{\mathsf{ek}_i}(T_i) \\
\sigma_{T_i} &= \mathrm{SPSign}(\mathsf{sk}_{\mathsf{Cl}}, T_i) \\
C_{\sigma_{T_i}} &= \mathrm{Com}(\sigma_{T_i}, _) \\
\pi_{E_{T_i}} &= \textsc{nizk}\Big\{T_i, \sigma_{T_i}, \mathsf{masked_inv}, B_1, \ldots, B_{t_{\mathsf{PP}}-1} : \\
&\qquad E_{T_i} = \mathrm{Enc}_{\mathsf{ek}_i}(T_i) \\
&\qquad \wedge\, T_i = \mathsf{masked_inv} \cdot \prod_{j=1}^{t_{\mathsf{PP}}-1} {B_j}^{(i^j)} \\
&\qquad \wedge\, \mathrm{SPVerif}(\sigma_{T_i}, T_i, \mathsf{vk}_{\mathsf{FS}})\Big\},
\end{aligned}$$

pour $1 \leqslant i \leqslant n_{\mathsf{PP}}$. Le client envoie finalement $(C_{B_j})_j$ et $(E_{T_i}, C_{T_i}, C_{\sigma_{T_i}}, \pi_{E_{T_i}})_i$ au fournisseur, qui accepte le partage si et seulement si les preuves sont valides.

La figure 5.13 décrit les deux partages.

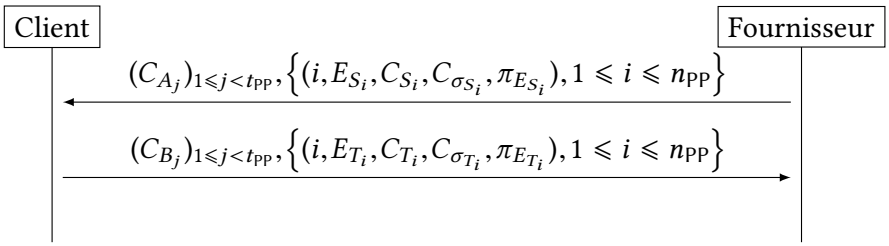


FIGURE 5.13 – Partage optimiste des secrets

Émission du témoignage

Scénario B – client malveillant Si le fournisseur ne reçoit pas le message m_1 du client, il demande aux porteurs de part de déchiffrer leur part en leur envoyant $\mathsf{id}_{\mathsf{trans}}$, $E_{T_i}, C_{\mathsf{masked\_inv}}$, $C_{T_i}, C_{\sigma_{T_i}}, \pi_{E_{T_i}}$ et $(B_j)_{1\leqslant j<t_{\mathsf{PP}}}$. Après avoir laissé le temps au client d'envoyer le message m_1 pour noter le fournisseur, chacun déchiffre sa part et prouve le déchiffrement :

$$\begin{aligned}
T_i &= \mathrm{Dec}_{\mathsf{dk}_{\mathsf{PP}_i}}(E_{T_i}) \\
\pi_{T_i} &= \textsc{nizk}\{\mathsf{dk}_{\mathsf{PP}_i} : \\
&\qquad T_i = \mathrm{Dec}_{\mathsf{dk}_{\mathsf{PP}_i}}(E_{T_i}) \\
&\qquad \wedge\, \mathsf{dk}_{\mathsf{PP}_i} \text{ est la clé de déchiffrement associée à } \mathsf{ek}_{\mathsf{PP}_i}\}.
\end{aligned}$$

Les porteurs de part envoient ensuite $T_i, C_{\mathsf{dk}_{\mathsf{PP}_i}}$ et π_{T_i} au fournisseur, qui peut alors reconstruire l'invariant masqué à partir de $t_{\mathsf{PP}} = \lceil n_{\mathsf{PP}}/2 \rceil$ parts, et calculer l'invariant :

$$\begin{aligned} \mathsf{masked\_inv} &= \mathrm{Interp}\left(\left\{(i_j, T_{i_j}),\ 1 \leqslant j \leqslant t_{\mathsf{PP}}\right\}\right) \\ \mathsf{inv} &= \mathrm{Unmask}(\mathsf{masked\_inv}, r_{\mathsf{pre\_inv}}). \end{aligned}$$

Finalement, le fournisseur peut émettre le témoignage à partir des éléments de la deuxième colonne de la table 5.5. La figure 5.14 présente cette interaction.

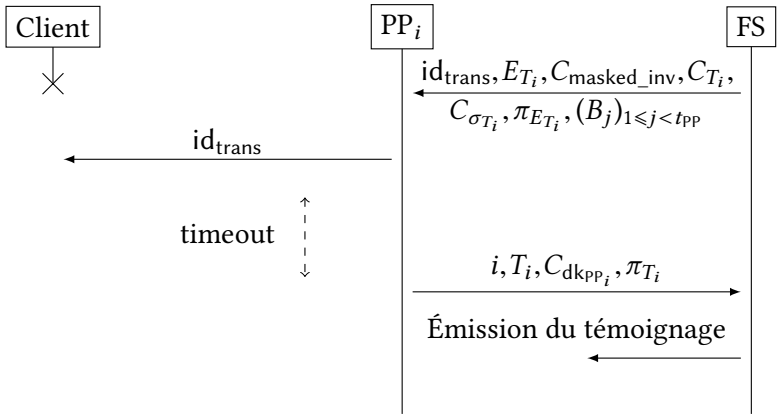


FIGURE 5.14 – Construction et émission du témoignage dans le scénario B, pour la variante optimiste

Scénario C – fournisseur malveillant Lorsque le fournisseur n'envoie pas le message m_2 au client, celui-ci demande les parts aux porteurs de part en leur envoyant $\mathsf{id}_{\mathsf{trans}}$, E_{S_i}, $C_{\mathsf{Id}_{\mathsf{FS}}}$, C_{S_i}, C_{σ_i}, $\pi_{E_{S_i}}$, $(A_j)_{1 \leqslant j < t_{\mathsf{PP}}}$. Après avoir vérifié la preuve, les porteurs de part signent la note du client, déchiffrent leur part et prouvent ces déchiffrements :

$$\begin{aligned} \sigma_{\rho,\mathsf{PP}_i} &= \mathrm{Sign}\left(\mathsf{sk}_{\mathsf{PP}_i}, H(\mathsf{id}_{\mathsf{trans}}, \rho)\right) \\ S_i &= \mathrm{Dec}_{\mathsf{dk}_{\mathsf{PP}_i}}(E_{S_i}) \\ \pi_{S_i} &= \textsc{nizk}\Big\{\mathsf{dk}_{\mathsf{PP}_i} : \\ &\qquad S_i = \mathrm{Dec}_{\mathsf{dk}_{\mathsf{PP}_i}}(E_{S_i}) \\ &\qquad \wedge\ \mathsf{dk}_{\mathsf{PP}_i} \text{ est la clé de déchiffrement associée à } \mathsf{ek}_{\mathsf{PP}_i}\Big\}. \end{aligned}$$

	Scénario A	Scénario B	Scénario C
FS	$\mathsf{Id}_{\mathsf{FS}}$, $\mathsf{vk}_{\mathsf{FS}}$, $\mathsf{nym}_{\mathsf{FS}}$, $C_{\mathsf{Id}_{\mathsf{FS}}}$, $\mathsf{cert}_{\mathsf{FS}}$, π_{FS}	$\mathsf{Id}_{\mathsf{FS}}$, $\mathsf{vk}_{\mathsf{FS}}$, $\mathsf{nym}_{\mathsf{FS}}$, $C_{\mathsf{Id}_{\mathsf{FS}}}$, $\mathsf{cert}_{\mathsf{FS}}$, π_{FS}	$\mathsf{Id}_{\mathsf{FS}}$, $\mathsf{nym}_{\mathsf{FS}}$, $C_{\mathsf{Id}_{\mathsf{FS}}}$, $C_{\mathsf{cert}_{\mathsf{FS}}}$, $\pi_{\mathsf{cert}_{\mathsf{FS}}}$, $\left\{(i, S_{i_j}, E_{S_{i_j}}, C_{S_{i_j}}, C_{\mathsf{dk}_{\mathsf{PP}_{i_j}}}, C_{\sigma_{\mathsf{FS},i_j}}, \pi_{E_{S_{i_j}}}, \pi_{S_{i_j}}), 1 \leqslant j \leqslant t_{\mathsf{PP}}\right\}$, $(C_{A_j})_{1 \leqslant j < t_{\mathsf{PP}}}$
Client	$\mathsf{nym}_{\mathsf{Cl}}$, $C_{\mathsf{id}_{\mathsf{Cl}}}$, $C_{\mathsf{cert}_{\mathsf{Cl}}}$, $\pi_{\mathsf{cert}_{\mathsf{Cl}}}$	$\mathsf{nym}_{\mathsf{Cl}}$, $C_{\mathsf{id}_{\mathsf{Cl}}}$, $C_{\mathsf{cert}_{\mathsf{Cl}}}$, $\pi_{\mathsf{cert}_{\mathsf{Cl}}}$	$\mathsf{nym}_{\mathsf{Cl}}$, $C_{\mathsf{id}_{\mathsf{Cl}}}$, $C_{\mathsf{cert}_{\mathsf{Cl}}}$, $\pi_{\mathsf{cert}_{\mathsf{Cl}}}$
Id. trans.	$\mathsf{id}_{\mathsf{trans}}, \varsigma_{\mathsf{FS}}, \varsigma_{\mathsf{Cl}}$	$\mathsf{id}_{\mathsf{trans}}, \varsigma_{\mathsf{FS}}, \varsigma_{\mathsf{Cl}}$	$\mathsf{id}_{\mathsf{trans}}, \varsigma_{\mathsf{FS}}, \varsigma_{\mathsf{Cl}}$
Invariant	pre_inv, masked_inv, inv, $C_{r_{\mathsf{masked\_inv}}}$, $\pi_{\mathsf{masked\_inv}}$	pre_inv, masked_inv, inv, $C_{\mathsf{masked\_inv}}$, $C_{r_{\mathsf{masked\_inv}}}$, $\pi_{C_{\mathsf{masked\_inv}}}$, $\left\{(i_j, T_{i_j}, E_{T_{i_j}}, C_{T_{i_j}}, C_{\sigma_{T_{i_j}}}, C_{\mathsf{dk}_{\mathsf{PP}_{i_j}}}, \pi_{E_{T_{i_j}}}, \pi_{T_{i_j}}), 1 \leqslant j \leqslant t_{\mathsf{PP}}\right\}$, $(C_{B_j})_{1 \leqslant j < t_{\mathsf{PP}}}$	inv, Π_{inv}
Note	ρ, ς_ρ, $\sigma_{\rho,\mathsf{FS}}$	N.A.	ρ, $(\sigma_{\rho,\mathsf{PP}_{i_j}})_j$

Table 5.5 – Éléments composant le témoignage pour la variante optimiste

Chaque porteur de part envoie ensuite S_i, $C_{\mathsf{dk}_{\mathsf{PP}_i}}$, $\sigma_{\rho,\mathsf{PP}_i}$ et π_{S_i} au client. Celui-ci peut alors reconstruire l'identifiant du fournisseur, puis calculer l'invariant et prouver son calcul :

$$\begin{aligned} \mathsf{Id}_{\mathsf{FS}} &= \mathrm{Interp}\left(\left\{(i_j, S_{i_j}),\ 1 \leqslant j \leqslant t_{\mathsf{PP}}\right\}\right) \\ \mathsf{inv} &= \mathrm{Inv}(\mathsf{Id}_{\mathsf{FS}}, \mathsf{id}_{\mathsf{Cl}}) \\ \pi_{\mathsf{inv}} &= \textsc{nizk}\left\{\mathsf{id}_{\mathsf{Cl}} : \mathsf{inv} = \mathrm{Inv}(\mathsf{Id}_{\mathsf{FS}}, \mathsf{id}_{\mathsf{Cl}})\right\}. \end{aligned}$$

Finalement, le client peut émettre le témoignage, qui comporte les éléments de la troisième colonne de la table 5.5. La figure 5.15 présente cette étape.

5.5 Bilan

Dans ce chapitre, nous avons présenté quatre mécanismes de réputation construits sur le même modèle et qui, comme nous le montrons au chapitre suivant, garantissent les propriétés de sécurité définies au chapitre 1 : l'indéniabilité des notes et des preuves de transaction, l'inforgeabilité des témoignages et des scores de réputation, et l'associabilité des témoignages. De plus, les clients peuvent émettre des notes positives ou négatives ; ces mécanismes permettent donc d'utiliser le moteur de réputation décrit en section 4.4 ainsi que la méthode de filtrage des témoignages injustes proposée par Whitby et coll. [WJI04], ce qui permet de se reposer sur un moteur de réputation précis et robuste aux attaques.

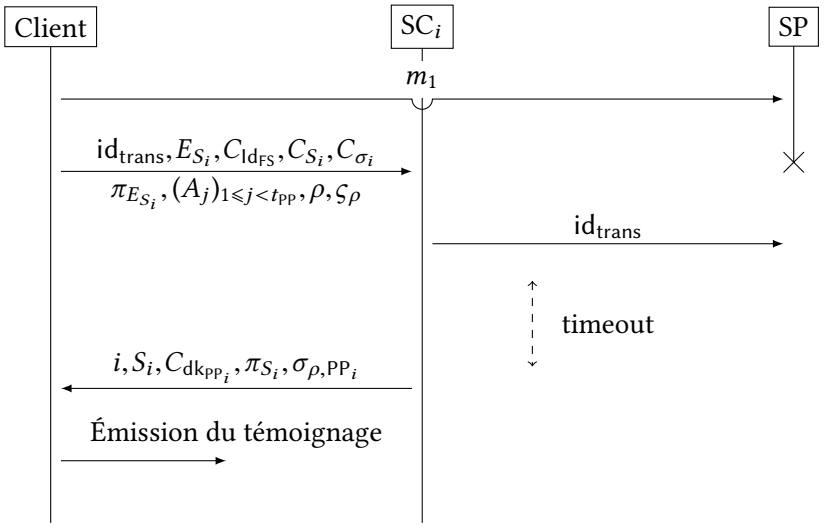


FIGURE 5.15 – Construction et émission du témoignage dans le scénario C, pour la variante optimiste

Les deux premiers mécanismes ne préservent pas la vie privée des fournisseurs ; ils améliorent néanmoins le mécanisme du chapitre 4 en associant les témoignages émis sur un même fournisseur (propriété 7) et en permettant aux clients de générer un nombre arbitraire de pseudonymes sans passer par une autorité centrale. Les deux derniers mécanismes préservent également la vie privée des fournisseurs, empêchant toute discrimination de la part des clients.

Dans les chapitres suivants, nous étudions plus précisément le mécanisme classique préservant la vie privée des clients et des fournisseurs : nous prouvons formellement qu'il respecte les propriétés de vie privée et de sécurité, et détaillons ses performances théoriques et expérimentales.

6 Preuves de vie privée et de sécurité

Nous montrons maintenant que le mécanisme classique préservant la vie privée des utilisateurs (voir section 5.4.2) répond à nos attentes, c'est-à-dire qu'il respecte toutes les propriétés énoncées au chapitre 1. Ces preuves reposent sur la sécurité des outils présentés au chapitre 2 ainsi que sur l'hypothèse SXDH.

Notre approche est la même qu'au chapitre 2 : pour prouver une propriété de sécurité, nous commençons par la formaliser à travers une expérience impliquant un adversaire $\mathcal{A}$. Pour simuler les capacités de cet adversaire, $\mathcal{A}$ a accès à des oracles ; plus précisément, nous souhaitons que l'adversaire puisse

1. faire interagir deux utilisateurs quelconques, à l'aide de l'oracle O_{inter} ;
2. corrompre un utilisateur, à l'aide de l'oracle O_{corr} ;
3. obtenir l'identifiant du client masqué dans l'invariant, à l'aide de l'oracle O_{inv}.

L'utilisation des oracles par l'adversaire est limitée suivant les propriétés prouvées ; par exemple, si l'objectif de $\mathcal{A}$ est de désanonymiser un utilisateur, il ne doit pas le corrompre.

Afin d'éviter de quelconques interférences entre les différents outils cryptographiques, leurs préparations doivent être indépendantes. Par exemple, nous avons utilisé un même G_1 pour les engagements SXDH, pour les signatures automorphes et pour l'invariant afin de simplifier les notations. En pratique, chacun de ces outils doit utiliser des éléments choisis indépendamment.

Dans ce chapitre, nous supposons que les tierces parties utilisées sont honnêtes, c'est-à-dire qu'au moins deux tiers des porteurs de part sont honnêtes, ainsi qu'une majorité des signataires accrédités. Nous avons expliqué en section 5.3.3 que le nombre de porteurs de part choisis pour une interaction peut être adapté pour rendre la probabilité d'une coalition négligeable ; dans la suite, nous notons $\mathrm{Av}_{\mathsf{PP}}(\kappa)$ l'avantage de l'adversaire pour contrôler suffisamment de porteurs de part, qui est un avantage négligeable. Nous discutons des propriétés lorsque les tierces parties sont corrompues en section 6.2. De plus, nous supposons que l'adversaire ne contrôle pas complètement le réseau : en effet, dans ce cas il peut facilement empêcher un client d'émettre une note en coupant ses connexions après une transaction. L'adversaire peut néanmoins écouter tous les messages échangés.

6.1 Preuve du mécanisme de réputation classique préservant la vie privée des clients et des fournisseurs

Nous prouvons maintenant que le mécanisme de réputation présenté en section 5.4.2 garantit les propriétés de vie privée et de sécurité définies au chapitre 1.

6.1.1 Vie privée des fournisseurs de service

La vie privée des fournisseurs de service (propriété 1) est préservée si l'expérience suivante renvoie 1 avec probabilité $1/2 + \mathrm{neg}(\kappa)$:

Vie privée des fournisseurs

1 : $\mathrm{Setup}(1^\kappa)$
2 : $(\mathsf{FS}_0, \mathsf{FS}_1) \leftarrow \mathcal{A}^{O_{\mathrm{corr}}(\cdot), O_{\mathrm{inter}}(\cdot,\cdot), O_{\mathrm{inv}}(\cdot)}()$
3 : $b \xleftarrow{R} \{0,1\}$
4 : $b' \leftarrow \mathcal{A}^{O_{\mathrm{corr}}(\cdot), O_{\mathrm{inter}}(\cdot,\cdot), O_{\mathrm{inv}}(\cdot)}(\mathsf{FS}_b, \mathsf{FS}_{\bar{b}})$
5 : L'expérience renvoie 1 si $b' = b$, et 0 sinon

Dans la deuxième étape de cette expérience, $\mathcal{A}$ peut faire interagir n'importe quels utilisateurs. $\mathcal{A}$ peut également corrompre n'importe quels utilisateurs. Les fournisseurs FS_0 et FS_1 qu'il choisit doivent avoir une réputation équivalente, et ne pas avoir été corrompus. Dans la quatrième étape, $\mathcal{A}$ ne peut pas corrompre FS_0, FS_1, FS_b ou $\mathsf{FS}_{\bar{b}}$. Finalement, $\mathcal{A}$ peut faire interagir les fournisseurs FS_b et $\mathsf{FS}_{\bar{b}}$ uniquement jusqu'à l'envoi du message m_1 par le client (voir section 5.4.2, page 109) – après cette étape, le fournisseur révèle son identité.

Démonstration. Pour chaque interaction où $\mathcal{A}$ fait interagir FS_b ou $\mathsf{FS}_{\bar{b}}$, il voit les éléments suivants :

$$\begin{aligned}
&\mathsf{nym}_{\mathsf{FS}}, C_{\mathsf{Id}_{\mathsf{FS}}}, C_{\mathsf{cert}_{\mathsf{FS}}}, \pi_{\mathsf{cert}_{\mathsf{FS}}},\\
&\mathsf{rep}_{\mathsf{FS}}, (C_{\sigma_{i_j},\mathsf{rep}})_{1\leqslant j\leqslant t_{\mathsf{SA}}}, \pi_{\mathsf{rep}},\\
&\mathsf{pre_inv}, \pi_{\mathsf{pre_inv}}, C_{r_{\mathsf{pre_inv}}},\\
&C_{\mathsf{PP}}, s_{\mathsf{PP}}, \varsigma_{\mathsf{FS}},\\
&(C_{A_j})_{1\leqslant j\leqslant t_{\mathsf{PP}}}, \{E_{S_i}, \pi_{S_i}\}_{1\leqslant i\leqslant n_{\mathsf{PP}}}\,.
\end{aligned}$$

Nous considérons n telles interactions, que nous modifions graduellement pour remplacer les éléments liés à FS_b par ceux de $\mathsf{FS}_{\bar{b}}$, et réciproquement. Nous montrons finalement que l'avantage de $\mathcal{A}$ pour se rendre compte d'une telle modification est négligeable.

Jeu 0 Le premier jeu considéré est celui où les interactions sont inchangées.

Jeu 1 Nous commençons par remplacer la crs du système de preuves NIZK par une crs simulée (voir section 2.5), ce qui permet de construire des preuves arbitraires avec la clé de trappe tk. Pour l'instant, les preuves concernent toujours des équations valides, et sont donc inchangées. L'avantage de l'adversaire pour distinguer ce jeu du jeu 0 est donc son avantage pour distinguer les crs, c'est-à-dire

$$\mathrm{Av}_{1/0}(\kappa) = \mathrm{Av}_{\mathrm{crs}}(\kappa).$$

Jeu 2 Cette étape vise à modifier les identifiants masqués dans les pré-invariants. Nous considérons pour cela la séquence hybride suivante, dans laquelle nous transformons progressivement les pré-invariants calculés à partir de $\mathsf{Id}_{\mathsf{FS}_b}$ en pré-invariants liés à $\mathsf{Id}_{\mathsf{FS}_{\bar{b}}}$; le jeu $\mathsf{Jeu}_{2,k}$ modifie le pré-invariant de la k-ième interaction, et conserve la validité des preuves grâce à la clé de trappe :

$\mathsf{Jeu}_{2,1}$	$\mathsf{Jeu}_{2,k}$	$\mathsf{Jeu}_{2,\mathsf{n}}$
pour $1 \leqslant i \leqslant n$,	pour $1 \leqslant i < k$,	pour $1 \leqslant i \leqslant n$,
pré-invariants avec $(\mathsf{FS}_b, \mathsf{FS}_{\bar{b}})$	pré-invariants avec $(\mathsf{FS}_{\bar{b}}, \mathsf{FS}_b)$	pré-invariants avec $(\mathsf{FS}_{\bar{b}}, \mathsf{FS}_b)$
	pour $k \leqslant i \leqslant n$,	
	pré-invariants avec $(\mathsf{FS}_b, \mathsf{FS}_{\bar{b}})$	

La seule différence entre deux jeux successifs est le pré-invariant d'une interaction, c'est-à-dire un chiffré Elgamal de l'identifiant du fournisseur. Comme l'algorithme de chiffrement Elgamal est sémantiquement sûr [GM84], l'avantage $\mathrm{Av}_{\mathrm{Elgamal}}(\kappa)$ pour qu'un adversaire distingue deux messages en connaissant uniquement leurs chiffrés est négligeable. C'est pourquoi l'avantage de $\mathcal{A}$ pour distinguer deux jeux successifs est $\mathrm{Av}_{\mathrm{Elgamal}}(\kappa)$. L'avantage total pour que $\mathcal{A}$ distingue les jeux 1 et 2 est donc

$$\mathrm{Av}_{2/1}(\kappa) = (n-1) \cdot \mathrm{Av}_{\mathrm{Elgamal}}(\kappa).$$

Jeu 3 Ce jeu vise à modifier les parts du partage de secret – les E_{S_i} – de sorte que le secret reconstruit, c'est-à-dire l'identifiant du fournisseur, soit permuté : $\mathsf{Id}_{\mathsf{FS}}$ doit être transformé en $\mathsf{Id}_{\mathsf{FS}_{\bar{b}}}$ et réciproquement. Considérons une interaction avec FS_b où les parts déchiffrées sont les $S_i = P(i)$ pour $1 \leqslant i \leqslant n$, où P est le polynôme dont les coefficients sont engagés dans $C_{\mathsf{Id}_{\mathsf{FS}_b}}$ et les C_{A_j}. Soit P' un polynôme de même degré que P, c'est-à-dire $(t_{\mathsf{PP}} - 1)$, tel que

$$\begin{cases} P'(0) = \mathsf{Id}_{\mathsf{FS}_{\bar{b}}} \\ P'(i) = P(i) \qquad \text{pour tout porteur de part } \mathsf{PP}_i \text{ corrompu par } \mathcal{A}. \end{cases}$$

Tant que l'adversaire a corrompu $(t_{\mathsf{PP}} - 1)$ ou moins porteurs de part – ce qui, comme expliqué en section 4.2, arrive avec une probabilité $(1 - \mathrm{Av}_{\mathsf{PP}}(\kappa))$ pour chaque interaction –, au plus t_{PP} points de ce polynôme sont fixés. Comme un polynôme de degré $(t_{\mathsf{PP}} - 1)$ est fixé par t_{PP} points, un polynôme P' existe avec une probabilité $(1 - \mathrm{Av}_{\mathsf{PP}}(\kappa))$. Nous notons $(A'_j)_{1 \leqslant j < t_{\mathsf{PP}}}$ ses coefficients.

Dans ce jeu, nous remplaçons les parts S_i par les $S'_i = P'(i)$; elles sont ensuite chiffrées pour obtenir les E'_{S_i}. De plus, en reprenant la sécurité sémantique de l'algorithme de chiffrement Elgamal, l'avantage de $\mathcal{A}$ pour se rendre compte de la modification d'une part est $\mathrm{Av}_{\mathrm{Elgamal}}(\kappa)$. Comme il y a au plus n_{PP} parts modifiées pour n interactions, l'avantage de $\mathcal{A}$ pour distinguer ce jeu du deuxième est

$$\mathrm{Av}_{3/2}(\kappa) \leqslant n \cdot n_{\mathsf{PP}} \cdot \mathrm{Av}_{\mathrm{Elgamal}}(\kappa) + n \cdot \mathrm{Av}_{\mathsf{PP}}(\kappa).$$

Rappelons que n_{PP} dépend de la sécurité désirée, et donc de κ (voir section 4.2) ; cependant, la figure 4.2b montre que n_{PP} est linéaire en κ.

Jeu 4 Nous modifions maintenant les engagements en remplaçant les éléments liés à FS_b pour les remplacer par ceux de $\mathsf{FS}_{\bar{b}}$, et inversement : nous remplaçons $\mathsf{nym}_{\mathsf{FS}_b}$ par $\mathsf{nym}_{\mathsf{FS}_{\bar{b}}}$, $C_{\mathsf{Id}_{\mathsf{FS}_b}}$ par $C_{\mathsf{Id}_{\mathsf{FS}_{\bar{b}}}}$, $C_{\mathsf{cert}_{\mathsf{FS}_b}}$ par $C_{\mathsf{cert}_{\mathsf{FS}_{\bar{b}}}}$, $C_{\sigma_{i_j,\mathsf{rep}_b}}$ par $C_{\sigma_{i_j,\mathsf{rep}_{\bar{b}}}}$, les engagements contenus dans $\varsigma_{\mathsf{FS}_b}$ par des $\varsigma_{\mathsf{FS}_{\bar{b}}}$, et les $(C_{A_j})_j$ par les $(C_{A'_j})$. Nous ajustons également les preuves afin qu'elles continuent à être valides. Comme, avec la crs simulée, le schéma d'engagement SXDH est parfaitement masquant,

$$\mathrm{Av}_{4/3}(\kappa) = 0.$$

Jeu 5 Maintenant que les engagements ont été modifiés, toutes les références à FS_b ont été remplacées par des références à $\mathsf{FS}_{\bar{b}}$, et réciproquement. De plus, les preuves portent à nouveau sur des équations valides ; la crs simulée et la clé de trappe ne sont donc plus nécessaires pour les construire. Nous pouvons alors repasser à une crs classique. L'avantage de l'adversaire est alors

$$\mathrm{Av}_{5/4}(\kappa) = \mathrm{Av}_{\mathsf{crs}}(\kappa).$$

Finalement, l'avantage de l'adversaire pour distinguer les interactions avec $(\mathsf{FS}_b, \mathsf{FS}_{\bar{b}})$ de celles impliquant $(\mathsf{FS}_{\bar{b}}, \mathsf{FS}_b)$ est

$$\begin{aligned}\mathrm{Av}_{\mathsf{VP\ FS}}(\kappa, n) &= \mathrm{Av}_{1/0}(\kappa) + \mathrm{Av}_{2/1}(\kappa) + \mathrm{Av}_{3/2}(\kappa) + \mathrm{Av}_{4/3}(\kappa) + \mathrm{Av}_{5/4}(\kappa) \\ &\leqslant 2 \cdot \mathrm{Av}_{\mathsf{crs}}(\kappa) + (n \cdot n_{\mathsf{PP}} + n + 1) \cdot \mathrm{Av}_{\mathrm{Elgamal}}(\kappa) + n \cdot \mathrm{Av}_{\mathsf{PP}}(-\kappa),\end{aligned}$$

où n est le nombre d'interactions observées par l'adversaire ; cet avantage est effectivement négligeable □

6.1.2 Vie privée des clients

La vie privée des clients (propriété 2) peut être divisée en deux parties. Tout d'abord, pendant une transaction, le fournisseur ne sait pas avec quel client il interagit – cette notion est proche de la propriété de vie privée des fournisseurs. De plus, les transactions d'un client avec différents fournisseurs ne sont pas associables. L'expérience suivante capture la première notion. Cette propriété est garantie si l'expérience renvoie 1 avec probabilité $1/2 + \mathrm{neg}(\kappa)$:

Vie privée des clients – indistinguabilité pré-transaction

1 : $\mathrm{Setup}(1^\kappa)$
2 : $(\mathsf{Cl}_0, \mathsf{Cl}_1) \leftarrow \mathcal{A}^{O_{\mathrm{corr}}(\cdot), O_{\mathrm{inter}}(\cdot,\cdot), O_{\mathrm{inv}}(\cdot)}()$
3 : $b \xleftarrow{R} \{0,1\}$
4 : $b' \leftarrow \mathcal{A}^{O_{\mathrm{corr}}(\cdot), O_{\mathrm{inter}}(\cdot,\cdot), O_{\mathrm{inv}}(\cdot)}(\mathsf{Cl}_b, \mathsf{Cl}_{\bar{b}})$
5 : L'expérience renvoie 1 si $b' = b$, et 0 sinon

Là encore, $\mathcal{A}$ peut corrompre et faire interagir n'importe quels utilisateurs. Les clients choisis Cl_0 et Cl_1 ne doivent pas avoir été corrompus, et ne peuvent pas être corrompus à la quatrième étape. Finalement, les clients Cl_b et $\mathsf{Cl}_{\bar{b}}$ ne doivent pas être corrompus, et peuvent interagir uniquement jusqu'à l'envoi du message m_1 – strictement avant ce message.

L'avantage de l'adversaire pour cette expérience peut être obtenu de la même manière que $\mathrm{Av}_{\mathrm{VP\ FS}}(\kappa, n)$; en effet, les éléments envoyés par le client sont similaires à ceux qui étaient envoyés par le fournisseur :

$$\begin{aligned}&\mathsf{nym}_{\mathsf{Cl}}, C_{\mathsf{id}_{\mathsf{Cl}}}, C_{\mathsf{cert}_{\mathsf{Cl}}}, \pi_{\mathsf{cert}_{\mathsf{Cl}}},\\ &C_{\mathsf{masked\_inv}}, \pi_{C_{\mathsf{masked\_inv}}}, C_{r_{\mathsf{masked\_inv}}},\\ &r_{\mathsf{PP}}, \varsigma_{\mathsf{Cl}},\\ &(C_{B_j})_{1\leqslant j\leqslant t_{\mathsf{PP}}}, \{E_{T_i}, \pi_{T_i}\}_{1\leqslant i\leqslant n_{\mathsf{PP}}}.\end{aligned}$$

La seule différence est que le fournisseur envoie directement pre_inv, tandis que le client envoie un engagement sur masked_inv, qui est parfaitement masquant avec la crs simulée. C'est pourquoi l'avantage de l'adversaire pour cette expérience est

$$\boxed{\mathrm{Av}_{\mathrm{VP\ Cl},1}(\kappa, n) \leqslant 2 \cdot \mathrm{Av}_{\mathsf{crs}}(\kappa) + n \cdot n_{\mathsf{PP}} \cdot \mathrm{Av}_{\mathrm{Elgamal}}(\kappa) + n \cdot \mathrm{Av}_{\mathsf{PP}}(\kappa),}$$

où n est le nombre d'interactions observées par $\mathcal{A}$.

Vie privée des clients – indistinguabilité à long terme

1 : Setup(1^κ)

2 : $\left(n, \mathsf{Cl}_0, \mathsf{Cl}_1, (\mathsf{FS}_i)_{1\leqslant i\leqslant n}\right) \leftarrow \mathcal{A}^{O_{\mathsf{corr}}(\cdot), O_{\mathsf{inter}}(\cdot,\cdot), O_{\mathsf{inv}}(\cdot)}()$

3 : $b \xleftarrow{\mathrm{R}} \{0,1\}$

4 : $b' \leftarrow \mathcal{A}^{O_{\mathsf{corr}}(\cdot), O_{\mathsf{inter}}(\cdot,\cdot), O_{\mathsf{inv}}(\cdot)}\left(\mathsf{Cl}_b, \mathsf{Cl}_{\bar{b}}, (\mathsf{FS}_i)_{1\leqslant i\leqslant n}\right)$

5 : L'expérience renvoie 1 si $b' = b$, et 0 sinon

Dans la deuxième étape de cette expérience, Cl_0 et Cl_1 sont deux clients n'ayant pas été corrompus par $\mathcal{A}$. Dans la quatrième étape, $\mathcal{A}$ ne peut pas corrompre Cl_0, Cl_1, Cl_b ou $\mathsf{Cl}_{\bar{b}}$. De plus, lorsque nous nous intéressons à l'associabilité des transactions d'un client avec différents fournisseurs, $\mathcal{A}$ ne peut pas faire interagir n'importe quels clients avec n'importe quels fournisseurs : il faut que

$$\left\{\mathsf{FS} \mid O_{\mathsf{inter}}(\mathsf{FS}, \mathsf{Cl}_0) \text{ ou } O_{\mathsf{inter}}(\mathsf{FS}, \mathsf{Cl}_1)\right\} \cap \left\{\mathsf{FS} \mid O_{\mathsf{inter}}(\mathsf{FS}, \mathsf{Cl}_b) \text{ ou } O_{\mathsf{inter}}(\mathsf{FS}, \mathsf{Cl}_{\bar{b}})\right\} = \emptyset,$$

c'est-à-dire qu'un fournisseur ayant interagi avec Cl_0 ou Cl_1 ne doit pas avoir également interagi avec Cl_b ou $\mathsf{Cl}_{\bar{b}}$, sans quoi l'adversaire peut utiliser l'invariant comme distingueur.

Démonstration. Dans cette preuve, nous considérons que les n interactions se déroulent suivant le scénario A, c'est-à-dire que nous considérons que les clients et fournisseurs sont honnêtes ; nous justifions ensuite que les scénarios B et C sont similaires. Dans ces interactions, les clients sont Cl_b ou $\mathsf{Cl}_{\bar{b}}$. Nous modifions graduellement ces interactions pour transformer les Cl_b en $\mathsf{Cl}_{\bar{b}}$, et réciproquement. Les éléments dépendant du client sont les suivants :

$$\begin{aligned}&\mathsf{nym}_{\mathsf{Cl}}, C_{\mathsf{id}_{\mathsf{Cl}}}, C_{\mathsf{cert}_{\mathsf{Cl}}}, \pi_{\mathsf{cert}_{\mathsf{Cl}}},\\ &C_{\mathsf{masked\_inv}}, \pi_{C_{\mathsf{masked\_inv}}}, C_{r_{\mathsf{masked\_inv}}},\\ &r_{\mathsf{PP}}, \varsigma_{\mathsf{Cl}},\\ &(C_{B_j})_{1\leqslant j\leqslant t_{\mathsf{PP}}}, \{E_{T_i}, \pi_{T_i}\}_{1\leqslant i\leqslant n_{\mathsf{PP}}}.\end{aligned}$$

Jeu 0 Le premier jeu considéré correspond aux interactions, inchangées.

Jeu 1 Nous commençons par remplacer la crs par une crs simulée, ce qui nous donne une clé de trappe tk grâce à laquelle nous pouvons construire des preuves NIZK sur des équations arbitraires. L'avantage de l'adversaire pour distinguer les jeux 0 et 1 est

$$\mathrm{Av}_{1/0}(\kappa) = \mathrm{Av}_{\mathsf{crs}}(\kappa).$$

Jeu 2 Ce jeu vise à modifier les invariants masqués et les invariants. Grâce à la clé de trappe, nous pouvons toujours construire des preuves valides.

Remarque
Les invariants masqués et les invariants sont reliés par un aléa choisi par le fournisseur, qui est divulgué dans π_{FS} :

$$\mathsf{inv} = \mathrm{Unmask}(\mathsf{masked\_inv}, r_{\mathsf{pre\_inv}}).$$

Ainsi, il faut modifier les deux simultanément. En outre, l'invariant masqué ne donne aucune information supplémentaire par rapport à l'invariant ; en effet,

$$\mathsf{masked\_inv} = \left(G_1{}^{r_{\mathsf{masked\_inv}}} \cdot Y_1{}^{\mathsf{id_{Cl}}}, \mathsf{inv} \cdot G_1{}^{r_{\mathsf{pre\_inv}} r_{\mathsf{masked\_inv}}} \cdot Y_1{}^{r_{\mathsf{pre\_inv}} \mathsf{id_{Cl}}}\right);$$

$\mathsf{masked\_inv_1}$ suit la distribution de $G_1{}^{r_{\mathsf{masked\_inv}}}$; comme G_1 est un générateur de $\mathbb{G}_1$ et que $r_{\mathsf{masked\_inv}}$ est choisi aléatoirement de manière uniforme, $\mathsf{masked\_inv_1}$ paraît aléatoire. De plus, $\mathsf{masked\_inv_2} = \mathsf{inv} \cdot \mathsf{masked\_inv_1}{}^{r_{\mathsf{pre\_inv}}}$; sa distribution suit donc celle de $\mathsf{masked\_inv_1}{}^{r_{\mathsf{pre\_inv}}}$, qui est également uniformément aléatoire. ■

L'avantage de l'adversaire pour distinguer les jeux 1 et 2 est donc son avantage pour distinguer $\mathrm{Inv}(\mathsf{Id_{FS}}_i, \mathsf{id_{Cl}}_b)$ de $\mathrm{Inv}(\mathsf{Id_{FS}}_i, \mathsf{id_{Cl}}_{\bar{b}})$ pour tout i, c'est-à-dire pour distinguer les deux jeux suivants – les invariants masqués et les preuves sont modifiés en même temps que les invariants. [1]

$\mathsf{Jeu}_{2,0}$	$\mathsf{Jeu}_{2,1}$
$\mathrm{Inv}(\mathsf{FS}_1,\mathsf{Cl}_b)$, $\mathrm{Inv}(\mathsf{FS}_1,\mathsf{Cl}_{\bar{b}})$	$\mathrm{Inv}(\mathsf{FS}_1,\mathsf{Cl}_{\bar{b}})$, $\mathrm{Inv}(\mathsf{FS}_1,\mathsf{Cl}_b)$
⋮	⋮
$\mathrm{Inv}(\mathsf{FS}_n,\mathsf{Cl}_b)$, $\mathrm{Inv}(\mathsf{FS}_n,\mathsf{Cl}_{\bar{b}})$	$\mathrm{Inv}(\mathsf{FS}_n,\mathsf{Cl}_{\bar{b}})$, $\mathrm{Inv}(\mathsf{FS}_n,\mathsf{Cl}_b)$

À cet effet, nous considérons la séquence hybride suivante, pour $1 \leqslant k \leqslant n$, où les différences entre jeux successifs sont surlignées en gris :

1. Par souci de simplification, nous utilisons FS_i au lieu de $\mathsf{Id_{FS}}_i$ dans la suite, et de même pour les identifiants des clients.

$\widetilde{\text{Jeu}}^{k}_{2,1}$	$\widetilde{\text{Jeu}}^{k}_{2,2}$	$\widetilde{\text{Jeu}}^{k}_{2,3}$	$\widetilde{\text{Jeu}}^{k}_{2,4}$
$\text{Inv}(\text{FS}_1,\text{Cl}_{\bar{b}}), \text{Inv}(\text{FS}_1,\text{Cl}_b)$	$\text{Inv}(\text{FS}_1,\text{Cl}_{\bar{b}}), \text{Inv}(\text{FS}_1,\text{Cl}_b)$	$\text{Inv}(\text{FS}_1,\text{Cl}_{\bar{b}}), \text{Inv}(\text{FS}_1,\text{Cl}_b)$	$\text{Inv}(\text{FS}_1,\text{Cl}_{\bar{b}}), \text{Inv}(\text{FS}_1,\text{Cl}_b)$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\text{Inv}(\text{FS}_{k-1},\text{Cl}_{\bar{b}}), \text{Inv}(\text{FS}_{k-1},\text{Cl}_b)$	$\text{Inv}(\text{FS}_1,\text{Cl}_{\bar{b}}), \text{Inv}(\text{FS}_1,\text{Cl}_b)$	$\text{Inv}(\text{FS}_1,\text{Cl}_{\bar{b}}), \text{Inv}(\text{FS}_1,\text{Cl}_b)$	$\text{Inv}(\text{FS}_1,\text{Cl}_{\bar{b}}), \text{Inv}(\text{FS}_1,\text{Cl}_b)$
$\text{Inv}(\text{FS}_k,\text{Cl}_b), \text{Inv}(\text{FS}_k,\text{Cl}_{\bar{b}})$	$R \xleftarrow{R} \mathbb{G}_1, \text{Inv}(\text{FS}_1,\text{Cl}_{\bar{b}})$	$R \xleftarrow{R} \mathbb{G}_1, R' \xleftarrow{R} \mathbb{G}_1$	$\text{Inv}(\text{FS}_1,\text{Cl}_{\bar{b}}), R' \xleftarrow{R} \mathbb{G}_1$
$\text{Inv}(\text{FS}_{k+1},\text{Cl}_b), \text{Inv}(\text{FS}_{k+1},\text{Cl}_{\bar{b}})$	$\text{Inv}(\text{FS}_1,\text{Cl}_b), \text{Inv}(\text{FS}_1,\text{Cl}_{\bar{b}})$	$\text{Inv}(\text{FS}_1,\text{Cl}_b), \text{Inv}(\text{FS}_1,\text{Cl}_{\bar{b}})$	$\text{Inv}(\text{FS}_1,\text{Cl}_b), \text{Inv}(\text{FS}_1,\text{Cl}_{\bar{b}})$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\text{Inv}(\text{FS}_n,\text{Cl}_b), \text{Inv}(\text{FS}_n,\text{Cl}_{\bar{b}})$	$\text{Inv}(\text{FS}_1,\text{Cl}_b), \text{Inv}(\text{FS}_1,\text{Cl}_{\bar{b}})$	$\text{Inv}(\text{FS}_1,\text{Cl}_b), \text{Inv}(\text{FS}_1,\text{Cl}_{\bar{b}})$	$\text{Inv}(\text{FS}_1,\text{Cl}_b), \text{Inv}(\text{FS}_1,\text{Cl}_{\bar{b}})$

Tout d'abord, notons que $\text{Jeu}_{2,0} = \widetilde{\text{Jeu}}^{1}_{2,1}$ et que $\text{Jeu}_{2,1} = \widetilde{\text{Jeu}}^{n+1}_{2,1}$. Nous allons montrer que, pour tout k, l'avantage de l'adversaire pour distinguer deux jeux successifs est négligeable. L'avantage de l'adversaire pour distinguer $\text{Jeu}_{2,0}$ et $\text{Jeu}_{2,1}$ sera alors lui aussi négligeable.

La seule différence entre les jeux $\widetilde{\text{Jeu}}^{k}_{2,1}$ et $\widetilde{\text{Jeu}}^{k}_{2,2}$ est que $\text{Inv}(\text{FS}_k,\text{Cl}_b)$ est remplacé par un aléa $R \xleftarrow{R} \mathbb{G}_1$. Soit $(G_1, {G_1}^a, {G_1}^b)$ un triplet DDH. Alors, en notant $a = \text{id}_{\text{Cl}_b}$ et $b = \log_{G_1}(\text{Id}_{\text{FS}_k})$, l'hypothèse DDH énonce que ${G_1}^{ab} = {\text{Id}_{\text{FS}_k}}^{\text{id}_{\text{Cl}_b}}$ est indistinguable d'un aléa, c'est-à-dire que $\text{Inv}(\text{FS}_k,\text{Cl}_b)$ est indistinguable de R. C'est-à-dire que l'avantage de l'adversaire pour distinguer ces deux jeux est $\text{Av}_{\text{DDH}}(\kappa)$, qui est négligeable.

Un raisonnement similaire permet de montrer que les jeux sont successivement indistinguables pour l'adversaire. Ainsi, au bout de $4 \cdot (n-1)$ transitions, l'avantage de l'adversaire pour distinguer $\text{Jeu}_{2,0}$ de $\text{Jeu}_{2,1}$ est au plus $4 \cdot (n-1)$ fois son avantage DDH. C'est pourquoi l'avantage de l'adversaire pour distinguer ces deux jeux – et donc les jeux 1 et 2 – est

$$\text{Av}_{2/1}(\kappa) = 4 \cdot (n-1) \cdot \text{Av}_{\text{DDH}}(\kappa).$$

Jeu 3 Lors de ce jeu, nous modifions tous les engagements et preuves associées pour passer de Cl_b à $\text{Cl}_{\bar{b}}$, similairement au jeu 4 de la vie privée des fournisseurs (voir section 6.1.1). Comme précédemment, l'avantage de l'adversaire pour distinguer les jeux 2 et 3 est nul :

$$\text{Av}_{3/2}(\kappa) = 0.$$

Jeu 4 Toutes les références à Cl_b ont été remplacées par des références à $\text{Cl}_{\bar{b}}$, et réciproquement. Les preuves portent également toutes sur des équations valides, c'est-à-dire que la crs simulée n'est plus nécessaire. Nous repassons donc à une crs classique. L'avantage de l'adversaire est alors

$$\text{Av}_{4/3}(\kappa) = \text{Av}_{\text{crs}}(\kappa).$$

Finalement, l'avantage de l'adversaire pour distinguer les interactions avec $(\text{Cl}_b, \text{Cl}_{\bar{b}})$ de celles impliquant $(\text{Cl}_{\bar{b}}, \text{Cl}_b)$ dans le scénario A est

$$\begin{aligned}\text{Av}_{\text{VP Cl,2,A}}(\kappa, n) &= \text{Av}_{1/0}(\kappa) + \text{Av}_{2/1}(\kappa) + \text{Av}_{3/2}(\kappa) + \text{Av}_{4/3}(\kappa) \\ &= 2 \cdot \text{Av}_{\text{crs}}(\kappa) + 4 \cdot (n-1) \cdot \text{Av}_{\text{DDH}}(\kappa).\end{aligned}$$

Dans le scénario B, c'est-à-dire lorsque le fournisseur est malveillant, il n'y a aucun élément supplémentaire dépendant du client ; l'avantage de l'adversaire est donc le même. Cependant, dans le scénario C, les parts sont issues de l'invariant masqué. Nous avons néanmoins expliqué que l'avantage de l'adversaire pour distinguer deux invariants masqués est nul ; l'avantage de l'adversaire est donc le même, c'est-à-dire

$$\boxed{\mathrm{Av}_{\mathrm{VP\ Cl},2}(\kappa, n) = 2 \cdot \mathrm{Av}_{\mathrm{crs}}(\kappa) + 4 \cdot (n-1) \cdot \mathrm{Av}_{\mathrm{DDH}}(\kappa),}$$

qui est négligeable. □

6.1.3 Indéniabilité des témoignages

L'indéniabilité des notes (propriété 3) capture la notion qu'une fois qu'une transaction a eu lieu, le fournisseur de service ne peut pas empêcher le client de le noter – et c'est bien ce fournisseur qui reçoit le témoignage. Cette propriété est garantie si le jeu suivant renvoie 1 avec probabilité $\mathrm{neg}(\kappa)$:

Indéniabilité des notes

1 : $\mathrm{Setup}(1^\kappa)$
2 : $(\mathsf{Cl}_0, \mathsf{FS}_0) \leftarrow \mathcal{A}^{O_{\mathrm{corr}}(\cdot), O_{\mathrm{inter}}(\cdot,\cdot), O_{\mathrm{inv}}}()$
3 : $\mathcal{A}$ fait interagir Cl_0 avec FS_0
4 : Cl_0 émet un témoignage tem_0
5 : L'expérience renvoie 1 si Cl_0 n'a pas été corrompu, et si tem_0 est invalide ou n'a pas été émis sur FS_0

Démonstration. Nous prouvons cette propriété en deux étapes : tout d'abord, nous montrons que le témoignage émis par le client est valide. Nous expliquons ensuite pourquoi c'est bien FS_0 qui reçoit ce témoignage.

Rappelons d'abord que, pour une crs classique, le système de preuves de Groth et Sahai est parfaitement sûr (voir section 2.5.1), c'est-à-dire que l'avantage de l'adversaire pour construire une preuve sur des équations invalides est nul. $\mathcal{A}$ ne peut donc pas tricher avant la transaction, et celle-ci se déroule seulement si le client a reçu les confirmations d'au moins $2 \cdot \lceil n_{\mathrm{PP}}/3 \rceil - 1$ porteurs de part. $\mathcal{A}$ peut mentir uniquement sur le message m_2 ; le client peut cependant détecter que la signature $\sigma_{\rho,\mathsf{FS}_0}$, le certificat $\mathsf{cert}_{\mathsf{FS}_0}$ ou les aléas de π_{FS_0} sont invalides. Comme le schéma d'engagement SXDH est parfaitement liant, l'avantage de l'adversaire pour forger un tel message est $\mathrm{Av}_{\mathrm{SP},\text{EU-CMA}}(\kappa, 1) + \mathrm{Av}_{\mathrm{SP},\text{EU-CMA}}(\kappa, 2)$, qui est négligeable. Que $\mathcal{A}$ envoie un message m_2 invalide ou n'en envoie pas, le résultat est le même : le client fait appel aux porteurs de part. De plus, le client a vérifié que $2 \cdot \lceil n_{\mathrm{PP}}/3 \rceil - 1$ porteurs de part ont confirmé avoir reçu une part telle que $S_i = Q(i)$, où Q est un polynôme tel que $Q(0) = \mathsf{Id}_{\mathsf{FS}_0}$. Pour empêcher le client de reconstruire $\mathsf{Id}_{\mathsf{FS}}$ et de pouvoir émettre sa note, $\mathcal{A}$ doit donc contrôler au moins $\lceil n_{\mathrm{PP}}/3 \rceil$ porteurs de part ; nous avons précédemment expliqué que son avantage pour y parvenir est $\mathrm{Av}_{\mathrm{PP}}(\kappa)$, qui est négligeable.

Ainsi, dans tous les cas, le client émet un témoignage valide. Montrons maintenant que ce témoignage est destiné à FS_0. Dans le scénario A, le fournisseur révèle $\mathsf{Id}_{\mathsf{FS}_0}$ et π_{FS} permet de vérifier l'engagement $C_{\mathsf{Id}_{\mathsf{FS}_0}}$; comme le schéma utilisé est parfaitement liant, $\mathcal{A}$ ne peut modifier

$\mathrm{Id}_{\mathrm{FS}_0}$. Dans le scénario C, le partage de secret vérifiable prouve également que le secret reconstruit correspond à l'engagement $C_{\mathrm{Id}_{\mathrm{FS}_0}}$. Comme le système de preuves est parfaitement sûr et que le schéma d'engagement est parfaitement liant, l'avantage de l'adversaire pour modifier le fournisseur est nul.

Finalement, l'avantage de l'adversaire pour réussir ce jeu est

$$\boxed{\mathrm{Av}_{\text{indéniab. notes}} = \mathrm{Av}_{\mathrm{SP},\text{EU-CMA}}(\kappa,1) + \mathrm{Av}_{\mathrm{SP},\text{EU-CMA}}(\kappa,2) + \mathrm{Av}_{\mathrm{PP}}(\kappa),}$$

qui est négligeable. □

L'indéniabilité des preuves de transactions (propriété 4) capture une notion similaire : les clients ne peuvent empêcher les fournisseurs d'obtenir une preuve de transaction, et celle-ci est bien associée au client. Cette propriété est garantie si l'expérience suivante renvoie 1 avec probabilité $\mathrm{neg}(\kappa)$:

Indéniabilité des preuves de transaction

1 : Setup(1^κ)
2 : $(\mathrm{Cl}_0, \mathrm{FS}_0) \leftarrow \mathcal{A}^{O_{\mathrm{corr}}(\cdot), O_{\mathrm{inter}}(\cdot,\cdot), O_{\mathrm{inv}}}()$
3 : $\mathcal{A}$ fait interagir Cl_0 avec FS_0
4 : FS_0 émet un témoignage tem_0
5 : L'expérience renvoie 1 si FS_0 n'a pas été corrompu, et si tem_0 est invalide ou que sa preuve de transaction n'est pas associée à Cl_0.

Un raisonnement similaire, reposant sur les engagements, les signatures, les preuves NIZK et la conformité du partage de secret, permet de prouver que l'avantage de l'adversaire pour l'indéniabilité des preuves de transactions est le même que pour l'indéniabilité des notes.

6.1.4 Inforgeabilité des témoignages

Un adversaire a deux options pour obtenir un témoignage illégitime : soit en corrompant suffisamment de signataires accrédités pour qu'ils acceptent ce témoignage comme valide, soit en forgeant un témoignage valide. Cependant, par hypothèse, le nombre de signataires accrédités corrompus est limité ; la première option est donc impossible.

Remarque
Un client et un fournisseur en collusion peuvent créer un témoignage sans effectuer de transaction ; il leur suffit de choisir les porteurs de part, de faire le partage des secrets et de leur transmettre le témoignage. Le jeu lié à cette propriété ne considère pas un tel témoignage comme une forge. De toute manière, si le client et le fournisseur émettent plusieurs témoignages de cette manière, le moteur de réputation peut exploiter l'associabilité des témoignages pour éviter les bourrages d'urne. ■

Le jeu suivant capture la notion d'inforgeabilité des témoignages (voir propriété 5). Cette propriété est garantie si ce jeu renvoie 1 avec probabilité $\mathrm{neg}(\kappa)$.

Inforgeabilité des témoignages

1 : $\mathrm{Setup}(1^\kappa)$

2 : $\mathsf{tem}_0 \leftarrow \mathcal{A}^{O_{\mathrm{corr}}(\cdot), O_{\mathrm{inter}}(\cdot,\cdot), O_{\mathrm{inv}}(\cdot)}()$

Renvoyer 1 si :

- tem_0, émis par $\mathsf{nym}_{\mathsf{Cl}_0}$ pour FS_0, est valide ;

3 : - $\mathsf{vk}_{\mathsf{Cl}_0} = \mathrm{Open}(\mathsf{nym}_{\mathsf{Cl}_0})$;[3]
- Cl_0 ou FS_0 n'a pas été corrompu par $\mathcal{A}$;
- tem_0 est différent de tout autre témoignage émis par Cl_0 sur FS_0.

Lorsque nous disons que « tem_0 est différent de tout autre témoignage émis par Cl_0 sur FS_0 », nous signifions que le témoignage tem_0 diffère de tout autre témoignage par au moins un des trois éléments finaux : l'identifiant de transaction $\mathsf{id}_{\mathsf{trans}0}$, la note ρ_0 ou l'invariant inv_0.

Démonstration. Tout d'abord, il y a six possibilités quant à la forge : les trois scénarios possibles d'émission d'un témoignage, et suivant si l'utilisateur non corrompu est le client ou le fournisseur. Nous montrons que l'avantage de l'adversaire pour forger un témoignage venant d'un client non corrompu dans le scénario A est négligeable, puis expliquons brièvement comment transposer ce raisonnement aux autres cas.

Pour montrer que l'adversaire ne peut modifier ni l'identifiant de transaction, ni l'invariant, ni la note du client, nous montrons dans un premier temps qu'il ne peut modifier ni l'identifiant de transaction ni la note du client en effectuant une réduction vers une forge de signature automorphe. Nous montrons ensuite que l'identifiant de transaction et l'invariant sont liés, et que l'adversaire ne peut donc modifier l'un sans l'autre.

Considérons un jeu EU-CMA pour le schéma de signature proxy anonyme (voir section 2.6.3). Nous obtenons une clé publique vk_0 et un oracle O_{Sign} qui permet de signer des messages avec la clé privée associée à vk_0 ; nous utilisons cette clé et l'oracle pour un client Cl_0. Nous montrons que si l'adversaire peut forger un nouveau témoignage, alors nous pouvons forger une signature sur un nouveau message, c'est-à-dire un message pour lequel nous n'avons pas demandé de signature via O_{Sign}. Comme le schéma de signature proxy anonyme est EU-CMA, cela signifie que les témoignages sont également inforgeables.

En plus de Cl_0, nous générons tous les autres utilisateurs du système, c'est-à-dire leurs clés, et les transmettons à l'adversaire. En supposant qu'il y ait n_{Cl} clients et que l'adversaire en corrompe n_{corr}, la probabilité pour qu'il ne corrompe pas Cl_0 et qu'il le choisisse est

$$P_{\mathsf{Cl}_0} = \underbrace{\left(\frac{n_{\mathsf{Cl}}-1}{n_{\mathsf{Cl}}} \cdot \dots \cdot \frac{n_{\mathsf{Cl}}-n_{\mathrm{corr}}}{n_{\mathsf{Cl}}-n_{\mathrm{corr}}+1}\right)}_{\mathsf{Cl}_0 \text{ n'est pas corrompu}} \cdot \underbrace{\left(\frac{1}{n_{\mathsf{Cl}}-n_{\mathrm{corr}}}\right)}_{\mathsf{Cl}_0 \text{ est choisi}} = \frac{1}{n_{\mathsf{Cl}}}.$$

Comme, à part pour Cl_0, nous connaissons les clés de tous les utilisateurs du système, nous pouvons simuler toutes les interactions n'impliquant pas Cl_0 ; l'oracle O_{Sign} nous permet de plus de simuler les interactions avec Cl_0. Les clients peuvent signer deux types de messages : pendant l'authentification mutuelle, les clients calculent $\varsigma_{\mathsf{Cl}_0}$, c'est-à-dire une signature sur $H(C_{\mathsf{PP}}, r_{\mathsf{PP}}, \mathsf{nym}_{\mathsf{FS}}, \mathsf{pre\_inv}, C_{\mathsf{masked\_inv}})$; pendant l'émission du témoignage, ς_{ρ_0}, c'est-à-dire une

signature sur $H(\mathsf{id}_{\mathsf{trans}}, \mathsf{masked\_inv}, \rho)$. Le premier type de signature est sur des messages qui sont le résultat d'une fonction de hachage de neuf éléments de $\mathbb{G}_1$, deux de $\mathbb{G}_2$ et un de $\mathbb{Z}_p$, tandis que le second type porte sur un message résultat d'une fonction de hachage de trois éléments de $\mathbb{G}_1$ et un de $\mathbb{Z}_p$; comme les messages sont structurellement différents et que la fonction de hachage utilisée est supposée résistante aux collisions, une signature du premier type ne peut être réutilisée pour le deuxième type. De plus, les deux types de messages comprennent des aléas choisis par le client et par le fournisseur, que nous supposons frais ; les attaques par rejeu [Syv94] sont donc impossibles.

Soit tem_0 le témoignage donné par l'adversaire à l'issue du jeu d'inforgeabilité, impliquant Cl_0 avec une probabilité $1/n_{\mathsf{Cl}}$ et qui est différent de tout autre témoignage émis par Cl_0 sur FS_0. Ce témoignage contient notamment ς_{ρ_0}, c'est-à-dire une signature proxy anonyme de Cl_0 sur $H(\mathsf{id}_{\mathsf{trans}0}, \mathsf{masked\_inv}_0, \rho_0)$. Comme ce témoignage est différent de tout autre témoignage émis par Cl_0 sur FS_0 si l'identifiant de transaction ou la note est différente, cette signature n'a pas été précédemment demandée à l'oracle. Dans ce cas, elle nous permet de forger une signature proxy anonyme.

Supposons maintenant que l'invariant est différent. Notons tout d'abord que si l'adversaire modifie le pré-invariant, nous pouvons encore forger une signature proxy anonyme : dans ς_{Cl}, le client signe en particulier le pré-invariant. De plus, la signature ς_{ρ_0} porte notamment sur $\mathsf{id}_{\mathsf{trans}0}$, qui dépend lui-même de $\mathsf{nym}_{\mathsf{FS}_0}$. De plus, π_{FS_0} comprend les aléas utilisés pour calculer $\mathsf{nym}_{\mathsf{FS}_0}$, qui le lient parfaitement à $\mathsf{vk}_{\mathsf{FS}_0}$; $\mathsf{vk}_{\mathsf{FS}_0}$ est de plus relié à $\mathsf{Id}_{\mathsf{FS}_0}$ par le certificat $\mathsf{cert}_{\mathsf{FS}_0}$, émis par l'autorité centrale. Finalement, $\mathsf{pre\_inv}_0$ – qui, comme expliqué précédemment, est signé par le client – et $\mathsf{Id}_{\mathsf{FS}_0}$ permettent de vérifier $r_{\mathsf{pre\_inv}_0}$ en vérifiant que

$$\mathsf{pre\_inv}_0 = \mathsf{Pre\_inv}(\mathsf{Id}_{\mathsf{FS}_0}, r_{\mathsf{pre\_inv}_0}).$$

Finalement, $r_{\mathsf{pre\_inv}_0}$ fait directement le lien entre $\mathsf{masked\_inv}_0$ et inv_0 ; or, $\mathsf{masked\_inv}_0$ fait partie du message signé par le client dans ς_{ρ_0}. Ainsi, si l'adversaire a modifié l'invariant, il a également modifié l'invariant masqué ou l'aléa utilisé pour le pré-invariant, et a donc forgé $\varsigma_{\mathsf{Cl}_0}$ ou ς_{ρ_0}. La figure 6.1 résume ces liens.

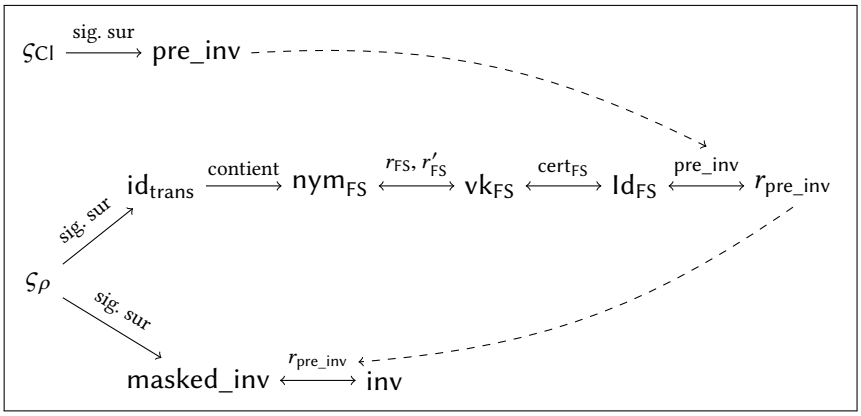


FIGURE 6.1 – Inforgeabilité d'un témoignage dans le scénario A, quand le client est honnête

L'avantage final de l'adversaire pour ce jeu, dans le scénario A et lorsque le client n'est pas

corrompu, est

$$\boxed{\mathrm{Av}_{\text{inforg. tém.,scén. A,Cl}}(\kappa) = 3 \cdot n_{\mathrm{Cl}} \cdot \mathrm{Av}_{\mathrm{AP},\text{EU-CMA}}(\kappa, 1),}$$

qui est négligeable. La figure 6.2 donne une idée de cette réduction.

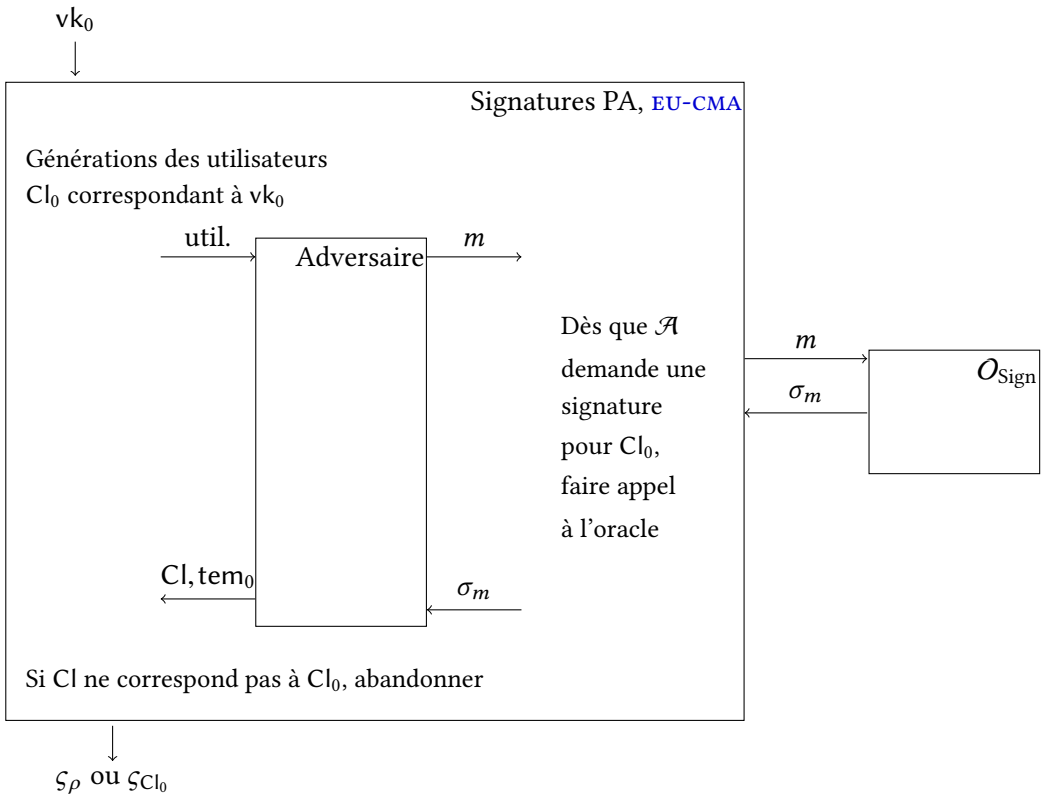


FIGURE 6.2 – Idée de la réduction pour l'inforgeabilité des témoignages

Les cinq autres cas sont traités similairement, et sont même parfois plus simples ; par exemple, la signature $\sigma_{\rho,\mathrm{FS}}$ suffit à garantir l'inforgeabilité des témoignages dans le scénario A lorsque le fournisseur est honnête puisqu'elle porte sur $H(\mathrm{id}_{\mathrm{trans}}, \sigma_\rho, \mathrm{inv})$. La figure 6.1 est également applicable à ces cas : les signatures du client doivent être remplacées par celles du fournisseur lorsque celui-ci est honnête, et les liens entre éléments sont modifiés lorsque les scénarios B et C sont considérés ; par exemple, dans le scénario C, le lien entre $\mathrm{nym}_{\mathrm{FS}}$ et $\mathrm{Id}_{\mathrm{FS}}$ passe par la preuve de certificat $\pi_{\mathrm{cert}_{\mathrm{FS}}}$ et par la reconstruction du secret grâce aux parts. □

6.1.5 Inforgeabilité des scores de réputation

L'inforgeabilité des scores de réputation (propriété 6) signifie qu'un fournisseur de service ne peut pas prouver un score de réputation autre que le sien. Cette propriété est garantie si le jeu suivant renvoie 1 avec probabilité $\mathrm{neg}(\kappa)$:

Inforgeabilité des scores de réputation

1 : $\mathsf{Setup}(1^\kappa)$

2 : $\left(C_{\mathsf{Id}_{\mathsf{FS}_0}}, \mathsf{rep}'_{\mathsf{FS}_0}, (C_{\sigma_{\mathsf{rep},ij}})_{1\leqslant j\leqslant t_{\mathsf{SA}}}, \pi_{\mathsf{rep}_0}\right) \leftarrow \mathcal{A}^{O_{\mathrm{corr}}(\cdot),O_{\mathrm{inter}}(\cdot,\cdot),O_{\mathrm{inv}}(\cdot)}()$
Renvoyer 1 si :

3 : — π_{rep_0} est valide pour $C_{\mathsf{Id}_{\mathsf{FS}_0}}$, $\mathsf{rep}'_{\mathsf{FS}_0}$ et $(C_{\sigma_{\mathsf{rep},ij}})_{1\leqslant j\leqslant t_{\mathsf{SA}}}$;
— $\mathsf{rep}_{\mathsf{FS}_0} \neq \mathsf{rep}'_{\mathsf{FS}_0}$.

Démonstration. Comme pour l'inforgeabilité des témoignages, nous considérons une réduction vers une forgeabilité face à des attaques par messages choisis pour le schéma de signature automorphe pour montrer que les scores de réputation sont également inforgeables.

À cet effet, nous obtenons une clé publique vk_0 et un oracle de signature O_{Sign} pour cette clé. Nous associons cette clé à un signataire accrédité SA_0, et générons les clés des autres utilisateurs du système. Nous pouvons donc simuler toutes les interactions, parfois en faisant appel à l'oracle pour SA_0.

Rappelons que l'adversaire contrôle au plus $t_{\mathsf{SA}} - 1$ signataires accrédités sur les t_{SA} nécessaires à la preuve de réputation d'un fournisseur. La probabilité pour que SA_0 ne soit pas corrompu par l'adversaire et soit utilisé dans la preuve de réputation est

$$P_{\mathsf{SA}_0} = \left(\frac{n_{\mathsf{SA}} - t_{\mathsf{SA}} + 1}{n_{\mathsf{SA}}}\right) \cdot \left(\frac{t_{\mathsf{SA}}}{n_{\mathsf{SA}}}\right) = \frac{t_{\mathsf{SA}} \cdot (n_{\mathsf{SA}} - t_{\mathsf{SA}} + 1)}{{n_{\mathsf{SA}}}^2}.$$

Soient FS_0 le fournisseur choisi par l'adversaire et $\left(C_{\mathsf{Id}_{\mathsf{FS}_0}}, \mathsf{rep}'_{\mathsf{FS}_0}, (C_{\sigma_{\mathsf{rep},ij}})_{1\leqslant j\leqslant t_{\mathsf{SA}}}, \pi_{\mathsf{rep}_0}\right)$ les éléments renvoyés par l'adversaire à l'issue du jeu d'inforgeabilité des scores de réputation. Supposons que ces éléments sont valides et que $\mathsf{rep}'_{\mathsf{FS}_0}$ est différente de la réputation de FS_0. Les seules signatures de SA_0 sont des signatures de réputation ; l'adversaire n'a pas pu utiliser de signature concernant un autre fournisseur puisque celles-ci portent notamment sur l'identifiant du fournisseur considéré ; de plus, l'adversaire n'a pas pu utiliser une signature antérieure du même fournisseur, puisque le round fait également partie du message signé. C'est pourquoi, si l'adversaire réussit ce jeu, nous n'avons pas fait appel à l'oracle O_{Sign} pour obtenir une signature de SA_0 ; comme de plus le schéma d'engagement est parfaitement liant, la signature engagée dans $C_{\sigma_{\mathsf{rep},0}}$ est une signature valide pour le jeu de forgeabilité du schéma de signature automorphe. Si l'adversaire réussit le jeu de forgeabilité des scores de réputation, nous pouvons donc réussir le jeu de forgeabilité du schéma de signature automorphe face à des attaques à messages choisis avec une probabilité P_{SA_0}. L'avantage de l'adversaire pour forger des signatures de réputation est donc

$$\boxed{\mathrm{Av}_{\text{forg. rép.}} = \frac{{n_{\mathsf{SA}}}^2}{t_{\mathsf{SA}} \cdot (n_{\mathsf{SA}} - t_{\mathsf{SA}} + 1)} \cdot \mathrm{Av}_{\mathrm{SP},\text{EU-CMA}},}$$

qui est effectivement négligeable. La figure 6.2 donne une idée de cette réduction. □

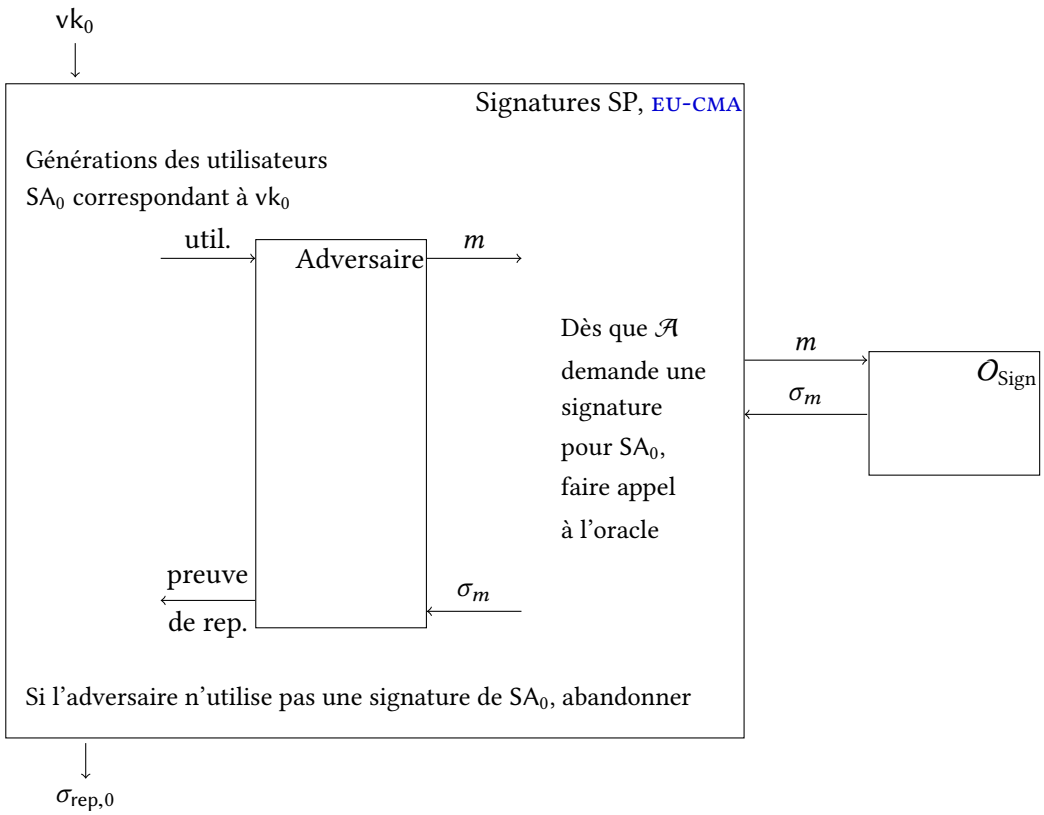


FIGURE 6.3 – Idée de la réduction pour l'inforgeabilité des scores de réputation

6.1.6 Associabilité des témoignages

L'associabilité des témoignages (propriété 7) assure que les témoignages d'un même client sur un même fournisseur sont associables. Cette propriété est garantie si le jeu suivant renvoie 1 avec probabilité $\mathrm{neg}(\kappa)$:

Associabilité des témoignages

1 : $\mathsf{Setup}(1^\kappa)$

2 : $(\mathsf{tem}_0, \mathsf{tem}_1) \leftarrow \mathcal{A}^{O_{\mathrm{corr}}(\cdot), O_{\mathrm{inter}}(\cdot,\cdot), O_{\mathrm{inv}}(\cdot)}$
Renvoyer 1 si :

- tem_0 et tem_1 sont valides ;
- tem_0 et tem_1 concernent tous deux FS ;

3 :
- En notant $\mathsf{nym}_{\mathrm{Cl}_0}$, $\mathsf{nym}_{\mathrm{Cl}_1}$ les pseudonymes des clients et inv_0, inv_1 les invariants associés à tem_0 et tem_1 :
  - $\mathrm{Open}(\mathsf{nym}_{\mathrm{Cl}_0}) = \mathrm{Open}(\mathsf{nym}_{\mathrm{Cl}_1})$ et $\mathsf{inv}_0 \neq \mathsf{inv}_1$, ou
  - $\mathrm{Open}(\mathsf{nym}_{\mathrm{Cl}_0}) \neq \mathrm{Open}(\mathsf{nym}_{\mathrm{Cl}_1})$ et $\mathsf{inv}_0 = \mathsf{inv}_1$.

Démonstration. Nous avons précédemment prouvé l'inforgeabilité des témoignages (voir section 6.1.4). Ainsi, si tem_0 (resp. tem_1) est valide, il contient $\mathsf{inv}_0 = \mathsf{Id}_{\mathsf{FS}}{}^{\mathsf{id}_{\mathsf{Cl}_0}}$ (resp. $\mathsf{inv}_1 = \mathsf{Id}_{\mathsf{FS}}{}^{\mathsf{id}_{\mathsf{Cl}_1}}$), où $\mathsf{vk}_{\mathsf{Cl}_0} = \mathsf{Open}(\mathsf{nym}_{\mathsf{Cl}_0})$ (resp. $\mathsf{vk}_{\mathsf{Cl}_1} = \mathsf{Open}(\mathsf{nym}_{\mathsf{Cl}_1})$). C'est pourquoi $\mathsf{inv}_0 = \mathsf{inv}_1$ si et seulement si $\mathsf{vk}_{\mathsf{Cl}_0} = \mathsf{vk}_{\mathsf{Cl}_1}$. Finalement, l'avantage de l'adversaire pour ce jeu est

$$\boxed{\mathrm{Av}_{\text{associabilité}}(\kappa) = 2 \cdot \mathrm{Av}_{\text{inforg. tém.}}(\kappa),}$$

qui est négligeable. □

6.2 Lorsque les tierces parties sont corrompues

Lorsque suffisamment de porteurs de part sont corrompus, le déroulement des interactions est compromis. Tout d'abord, si un client collabore avec eux, il peut apprendre l'identité des fournisseurs avec lesquels il interagit *avant* les transactions, grâce aux parts. Inversement, un fournisseur peut savoir s'il a déjà interagi avec ses clients avant la transaction. Cependant, deux fournisseurs ne peuvent toujours pas comparer leurs clients. Les porteurs de part peuvent également empêcher l'émission des témoignages, par exemple en ne transmettant pas leurs parts dans les scénarios B et C. Les porteurs de part corrompus ne peuvent toutefois pas compromettre les autres propriétés, c'est-à-dire l'inforgeabilité des témoignages et des scores de réputation, ainsi que l'associabilité des témoignages.

La compromission des signataires accrédités est plus grave. En effet, une majorité de signataires accrédités corrompus peut également dénier un témoignage – en ne le prenant pas en compte lors du calcul des scores de réputation. Plus grave, ils peuvent directement affecter une réputation arbitraire à chaque fournisseur ! Ainsi, même s'ils ne compromettent ni l'inforgeabilité des témoignages ou leur associabilité, ils peuvent inverser les effets du mécanisme de réputation en affectant une bonne réputation aux fournisseurs malveillants, et vice-versa. Malgré cela, les signataires accrédités ne peuvent pas compromettre la vie privée des utilisateurs.

6.3 Bilan

Dans ce chapitre, nous avons démontré que le mécanisme classique préservant la vie privée des utilisateurs – présenté en section 5.4.2 – garantit les propriétés de sécurité et de vie privée décrites au chapitre 1. Ces démonstrations se reposent principalement sur la sécurité des blocs cryptographiques utilisés – engagements SXDH, signatures automorphes et preuves NIZK –, présentés au chapitre 2. Nous avons supposé que les tierces parties utilisées – porteurs de part et signataires accrédités, pour ce système – étaient honnête. La section 4.2 décrit un choix des tierces parties réduisant la probabilité des collusions, ce qui justifie cette hypothèse.

Nous n'avons prouvé qu'un seul des quatre mécanismes présentés au chapitre 5. Les preuves présentées ici peuvent néanmoins être réutilisées : le fonctionnement des quatre mécanismes est similaire, et les avantages de l'adversaire pour chacune des propriétés sont du même ordre de grandeur.

7 Analyse des performances

Après avoir prouvé que les mécanismes de réputation présentés au chapitre 5 vérifient les propriétés introduites au chapitre 1, nous nous intéressons maintenant à leurs performances. En effet, les signatures de réputation de Bethencourt et coll. [BSS10] garantissent l'anonymat des clients et des fournisseurs, mais leur utilisation est limitée par leur taille : une signature de réputation sur 100 témoignages fait 50 Mio, ce qui représente un volume de données considérable (voir section 3.5.2). Nous montrons dans ce chapitre que nos quatre mécanismes sont efficaces ; leur utilisation est donc possible.

Nous commençons par étudier les performances théoriques de nos mécanismes, en utilisant les chiffres de Aranha et coll. [Ara+11], aussi bien en termes de taille des messages échangés qu'en temps de calcul. Nous présentons ensuite les performances d'une implémentation du mécanisme préservant la vie privée des clients et des fournisseurs et utilisant du partage de secret classique. Cette implémentation repose sur le framework Python Charm [Aki+13], qui facilite le prototypage de primitives cryptographiques complexes, comme les preuves NIZK ou les signatures proxy anonymes. Nous montrons que cette implémentation, bien que moins performante que sa version théorique, reste utilisable ; en particulier, des machines peu puissantes comme des Raspberry Pi pourraient être utilisées pour les porteurs de part.

Dans la suite, nous considérons un système large-échelle composé de $N = 10^8$ fournisseurs, dont $m = 5\,\%$ sont malveillants et en collusion. Comme motivé en section 4.2, nous choisissons $p_{\max} = 2^{-64}$ pour les mécanismes utilisant des gestionnaires de score, c'est-à-dire ceux présentés en section 5.2, ce qui nécessite 103 gestionnaires de score pour le mécanisme classique et 51 pour la variante optimiste. Pour les mécanismes présentés en section 5.4, nous choisissons $p_{\max} = 2^{-20}$ (voir section 5.3.3) ; dans ce cas, il faut 28 porteurs de part pour le mécanisme classique et 15 pour la variante optimiste. En plus de ne solliciter des utilisateurs externes que lorsqu'ils sont nécessaires, les mécanismes optimistes permettent donc de réduire de moitié leur nombre. Nous considérons également $n_{\text{SA}} = 10$ signataires accrédités. La table 7.1 résume ces valeurs.

Mécanisme	SA	GS / PP	
variante		classique	optimiste
Vie privée des clients	N.A.	103	51
Vie privée des utilisateurs	10	28	15

TABLE 7.1 – Nombre d'utilisateurs composant les tierces parties pour les quatre mécanismes, pour $N = 10^8$ et $m = 5\,\%$.

7.1 Étude théorique

Pendant une interaction, les utilisateurs calculent des preuves NIZK. La vérification d'une telle preuve nécessite de calculer des couplages (voir section 2.5), qui sont coûteux. Même en optimisant la vérification de ces preuves [Bla+10] (voir section 2.5.2), de nombreux couplages sont calculés. Par exemple, pour le mécanisme optimiste préservant la vie privée des clients, le fournisseur doit calculer 4 718 couplages avant l'interaction, c'est-à-dire pour l'authentification mutuelle et la vérification du partage. Optimiser leur calcul est donc nécessaire.

À cet effet, nous utilisons les travaux de Aranha et coll. [Ara+11] : les auteurs considèrent une courbe elliptique de la famille Barreto-Naehrig, où les éléments de $\mathbb{Z}_p$ et $\mathbb{G}_1$ sont représentés en 32 o, et les éléments de $\mathbb{G}_2$ en 64 o. En utilisant la méthode de calcul des auteurs, les quatre cœurs d'un processeur AMD Phenom II X4 940 à 3 GHz – un processeur haut de gamme de 2010 – calculent 8 couplages par milliseconde, 16 exponentiations dans $\mathbb{G}_2$ ou 48 dans $\mathbb{G}_1$.

Phase	Cl ↔ FS	FS ↔ GS	Cl ↔ GS	témoignage
Préparation	10,22	6,44	263,94	—
Scénario A	0	0	0	8,44
Scénario B	0	14,22	0	22,0

(a) pour le mécanisme classique

Phase	Cl ↔ FS	FS ↔ GS	Cl ↔ GS	témoignage
Préparation	126,88	0	0	—
Scénario A	0	0	0	8,44
Scénario B	0	124,32	0	86,59

(b) pour le mécanisme optimiste

TABLE 7.2 – Tailles des messages échangés pour le mécanisme sans anonymat du fournisseur, en kibioctets – les colonnes « Cl ↔ GS » et « Cl ↔ PP » représentent les échanges avec l'ensemble des gestionnaires de score ou des porteurs de part.

Les tables 7.2 et 7.3 présentent les tailles des messages échangés ainsi que du témoignage résultant d'une interaction dans les différents scénarios, pour les quatre mécanismes. Pour chaque système, les résultats sont satisfaisants : dans tous les cas, une interaction requiert quelques centaines de kibioctets, et un témoignage fait environ 10 Kio lorsque les utilisateurs sont honnêtes. Nous constatons également que, pour le mécanisme sans anonymat du fournisseur, la variante optimiste réduit considérablement la taille des messages échangés : la préparation de la variante classique demande 280,60 Kio, tandis que celle de la variante optimiste n'en demande que 126,88. Même lorsque le client est malveillant, la variante optimiste échange moins de données que le mécanisme classique : 251,2 Kio contre 294,82 Kio. Le seul inconvénient est qu'alors, le témoignage est quatre fois plus gros – sa taille reste en dessous de 100 Kio, alors que les signatures de réputation de Bethencourt demandent 500 Kio par témoignage (voir figure 3.5).

Phase	Cl ↔ FS	FS ↔ PP	Cl ↔ PP	témoignage
Preuve de réputation	47,75	0	0	—
Partage	3,28	32,38	60,38	—
Scénario A	2,94	0	0	12,06
Scénario B	0	7,5	0	19,63
Scénario C	0	0	11,26	21,38

(a) pour le mécanisme classique

Phase	Cl ↔ FS	FS ↔ PP	Cl ↔ PP	témoignage
Preuve de réputation	47,75	0	0	—
Partage	156,62	0	0	—
Scénario A	2,94	0	0	12,06
Scénario B	0	82,5	0	85,94
Scénario C	0	0	34,0	41,88

(b) pour le mécanisme optimiste

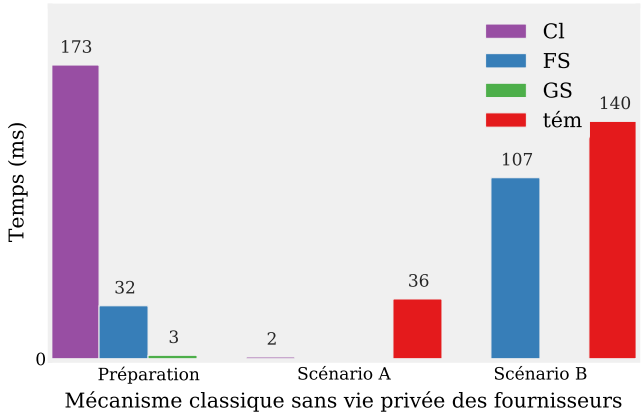
Table 7.3 – Tailles des messages échangés pour le mécanisme avec anonymat du fournisseur, en kibioctets – les colonnes « Cl ↔ GS » et « Cl ↔ PP » représentent les échanges avec l'ensemble des gestionnaires de score ou des porteurs de part.

Pour le mécanisme préservant la vie privée des clients et des fournisseurs, la variante optimiste requiert légèrement plus de ressources que le mécanisme classique : entre 200 Kio et 300 Kio contre 150 Kio. Néanmoins, la variante permet de ne faire appel aux porteurs de part que lorsqu'ils sont effectivement nécessaires. Pour ce mécanisme, les seuls messages dépendant du nombre de signataires accrédités, n_{SA}, est la preuve de réputation. Plus précisément, la preuve de réputation demande $6{,}44t_{\mathrm{SA}} + O(1)$ Kio. La taille des partages de secret a une complexité en $O({n_{\mathrm{GS}}}^2)$ – ou en $O({n_{\mathrm{PP}}}^2)$. Cependant, nous avons expliqué en Section 4.2 que le nombre de gestionnaires de score – ou porteurs de part – nécessaire est limité. Dans notre cas, il faut au maximum 103 gestionnaires de score pour assurer un niveau de sécurité de 2^{64} dans un système à très large échelle, ce qui reste limité. Finalement, les gestionnaires de score – ou les signataires accrédités – ne conservent que les témoignages finaux, c'est-à-dire au plus 80 Kio par interaction dans le pire cas. Notons qu'une fois un témoignage accepté par tous les signataires accrédités, ils n'ont plus besoin de conserver les preuves, ce qui limite les données stockées à une centaine d'octets par transaction.

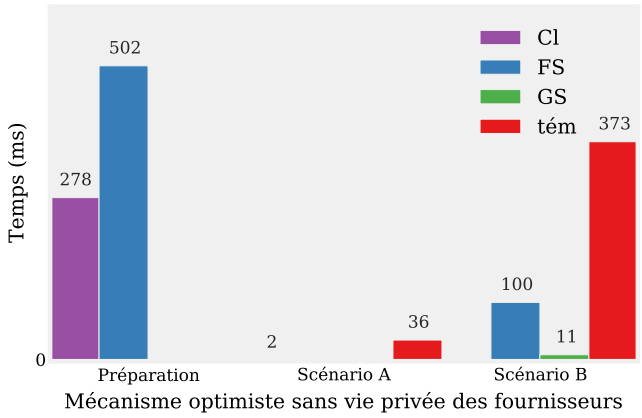
De manière générale, il y a moins de messages échangés lorsque les utilisateurs sont honnêtes. Pour minimiser le coût des mécanismes, il est possible soit d'inciter les utilisateurs à se comporter correctement – c'est-à-dire à ce que les clients émettent un témoignage et que les fournisseurs les aident à le faire –, où de pénaliser les utilisateurs malveillants, par exemple en les empêchant d'interagir pendant une période de temps donnée.

La figure 7.1 présente les temps de calcul théoriques de chaque utilisateur, ainsi que le temps

nécessaire à la vérification des témoignages, pour les deux mécanismes préservant la vie privée des clients, en millisecondes. Ces figures montrent des temps de calcul raisonnables. Pour le premier mécanisme classique, chaque utilisateur nécessite moins de 200 ms. De plus les gestionnaires de score, qui sont les seuls participants non directement impliqués dans l'interaction, ont seulement besoin de 3 ms pendant l'interaction, et de 140 ms pour vérifier le témoignage dans le pire cas ! La complexité du partage est également en $O({n_{\text{GS}}}^2)$ mais, comme expliqué précédemment, le nombre de gestionnaires de score passe à l'échelle. Notons que la variante optimiste de ce mécanisme a un coût, particulièrement pour les fournisseurs : leur préparation passe de 30 ms à 500 ms.



(a) pour le mécanisme classique

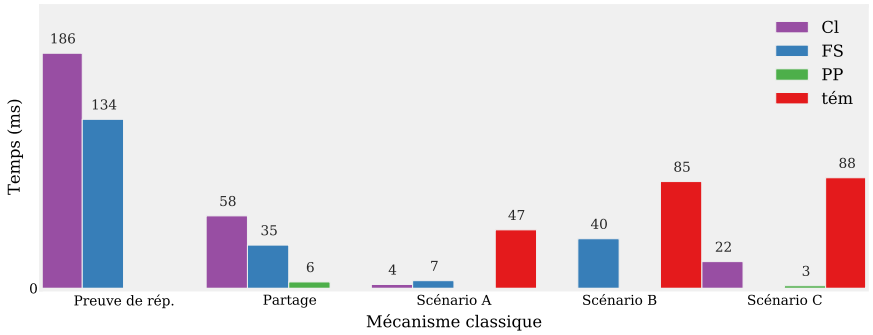


(b) pour le mécanisme optimiste

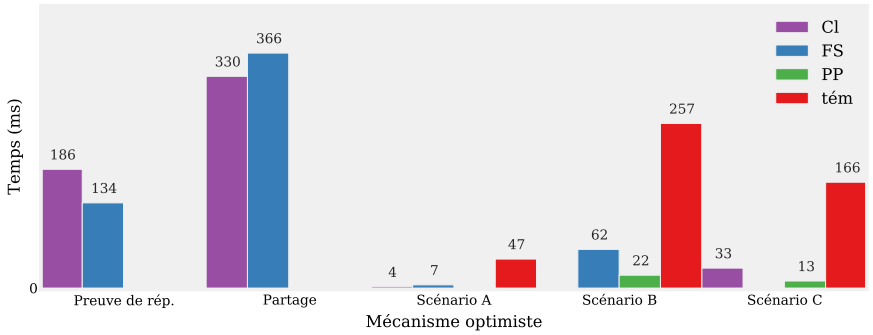
FIGURE 7.1 – Temps de calcul théoriques pour le mécanisme sans anonymat du fournisseur, en millisecondes

La figure 7.2 s'intéresse à ces temps pour les mécanismes préservant la vie privée des clients et des fournisseurs. Comme nous pouvons le voir, ces temps sont du même ordre de grandeur que précédemment : une interaction complète prend quelques centaines de millisecondes. Pour

la variante classique, le client et le fournisseur ont besoin de moins de 250 ms pour tous leurs calculs, tandis que les porteurs de part on besoin au plus de 10 ms lorsque le fournisseur est malveillant. Finalement, la vérification d'un témoignage demande moins de 100 ms. Lorsque la variante optimiste est utilisée, la préparation du client et du fournisseur prend environ 500 ms. Cependant, les porteurs de part ont besoin de moins de 20 ms lorsque le client ou le fournisseur est malveillant. Finalement, dans ces deux scénarios, la vérification du témoignage prend deux ou trois fois plus de temps, ce qui reste acceptable.



(a) pour le mécanisme classique



(b) pour le mécanisme optimiste

FIGURE 7.2 – Temps de calcul théoriques pour le mécanisme avec anonymat du fournisseur, en millisecondes

7.2 Implémentation du mécanisme de réputation

Nous avons implémenté le mécanisme de réputation en Python 2.7 à l'aide du framework Charm [Aki+13], ce qui a permis de prototyper simplement les primitives cryptographiques présentées au chapitre 2 et en section 5.1. De plus, Charm intègre un module de benchmark, ce qui permet de mesurer à la fois les temps de calcul et le nombre d'éléments échangés par les utilisateurs. Nous utilisons également Twisted [1], un framework réseau évènementiel, pour

1. https://twistedmatrix.com/trac/

gérer les communications entre les différents utilisateurs.

Les expériences ont été effectuées sur du matériel hétérogène, plus précisément dans une machine virtuelle sur un ordinateur portable Dell Latitude E6430 doté d'un processeur Core i7-3720 QM à 2,6 GHz, et sur des Raspberry Pi modèle B, versions 1 et 2.

La figure 7.3 montre les temps de calcul des utilisateurs pour le mécanisme classique préservant la vie privée des fournisseurs en secondes, sur la machine virtuelle. Plus précisément, cette figure montre les moyennes et écarts-types des temps de calcul sur dix simulations, en utilisant $n_{\mathrm{PP}} = 28$ porteurs de part et $n_{\mathrm{SA}} = 10$ signataires accrédités.

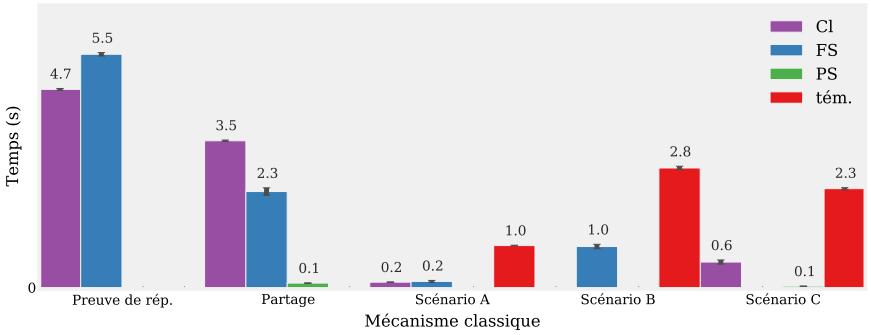


FIGURE 7.3 – Temps de calcul de l'implémentation du mécanisme classique préservant la vie privée des fournisseurs, en secondes

Cette figure montre que les temps de calcul de l'implémentation sont plus élevés que les temps théoriques, ce qui peut s'expliquer facilement. Tout d'abord, Aranha et coll. ont choisi avec soin une courbe de Barreto-Naehrig, sur laquelle ils ont optimisé le calcul des couplages en utilisant de l'assembleur et du code C [Ara+11]. Dans notre cas, nous utilisons la courbe MNT-159 proposée par Charm, qui est un framework Python autour de la bibliothèque pbc développée par Lynn [2]. De plus, les temps de calcul théoriques supposent que tous les calculs sont parallélisés, ce qui n'est pas le cas pour l'implémentation. Finalement, les calculs de tous les utilisateurs tournent sur la même machine virtuelle ; cela ne ralentit pas les phases où les calculs sont séquentiels, comme la preuve de réputation ou la construction du témoignage dans le scénario A, mais cela impacte les calculs qui pourraient être parallélisés, comme le partage des secrets. De manière générale, notre implémentation est environ 25 fois plus lente que les temps annoncés précédemment.

Malgré ces limitations, nous observons que notre mécanisme de réputation permet aux clients d'interagir avec des fournisseurs, et d'effectuer toute la préparation d'une interaction en 8 s. Les temps sont similaires pour les fournisseurs, ce qui leur permet d'interagir avec plusieurs clients simultanément. L'émission du témoignage peut prendre un peu de temps, notamment lorsque le client est malveillant.

Comme les temps de calcul des porteurs de part sont faibles, ils peuvent même utiliser un Raspberry Pi pour leurs calcul : avec la première version du modèle B, le partage des secrets prend seulement 4,7 s. Le temps d'attente des clients est alors augmenté, mais le délai reste

2. http://crypto.stanford.edu/pbc/

acceptable : la préparation prend toujours moins de 20 s, ce qui est bien moins long que le temps nécessaire à l'achat d'un bien sur un site de commerce électronique. La version 2 du Raspberry Pi modèle B permet un gain d'environ 33 % sur les temps de calcul, ce qui n'est pas négligeable.

Conclusion

Cette thèse présente le premier mécanisme de réputation préservant la vie privée des clients et des fournisseurs de service et permettant aux clients d'émettre des témoignages positifs et négatifs. Pour atteindre cet objectif, nous avons commencé par motiver un compromis raisonnable entre précision de la réputation et vie privée ; ce compromis permet de détecter – et donc de diminuer l'effet – des attaques comme le bourrage d'urne, tout en garantissant que les interactions des clients ne peuvent être associées à travers différents fournisseurs de service. Les définitions claires des propriétés désirées ont permis de comparer les mécanismes existants et de constater que, jusqu'à présent, ces mécanismes délaissaient la vie privée au profit de la précision de la réputation, ou inversement.

Nous avons ensuite présenté un mécanisme de réputation préservant la vie privée des clients ; la précision de la réputation n'est cependant pas optimale puisque l'associabilité des témoignages n'est pas garantie : plus les clients disposent de pseudonymes, plus les bourrages d'urnes sont faciles à mettre en œuvre. Néanmoins, ce mécanisme utilise des tierces parties distribuées pour garantir l'inforgeabilité des témoignages et des scores de réputation ainsi que l'indéniabilité des témoignages. De plus, ce mécanisme utilise un nouveau moteur de réputation, que nous montrons précis même en présence de témoins menteurs.

Après cela, nous avons amélioré ce mécanisme pour garantir l'associabilité des témoignages et pour permettre aux clients de générer eux-mêmes leurs pseudonymes, sans dépendre d'une autorité centrale ; à cet effet, nous utilisons de nouvelles constructions cryptographiques telles que l'invariant ou le partage de secret vérifiable optimiste. Finalement, nous modifions ce mécanisme en introduisant deux nouvelles tierces parties pour garantir les propriétés de sécurité tout en préservant la vie privée des clients et des fournisseurs de service.

Nous avons validé ces mécanismes de deux manières : tout d'abord, nous avons prouvé formellement que le mécanisme de réputation préservant la vie privée des clients et des fournisseurs garantit les propriétés de vie privée et de sécurité. Ensuite, nous avons analysé deux métriques du système : la taille des messages échangés, et les temps de calcul des utilisateurs. Une étude théorique préliminaire montre que les utilisateurs n'ont besoin que de quelques centaines de kibioctets et de quelques centaines de millisecondes pour leurs calculs ; une implémentation haut-niveau du mécanisme de réputation montre que les utilisateurs n'ont besoin que de quelques secondes pour leurs calculs, ce qui reste réaliste. De plus, les calculs des utilisateurs qui ne sont pas impliqués directement dans l'interaction – c'est-à-dire les porteurs de part – restent légers, et les porteurs de part peuvent même utiliser des ordinateurs bon marchés à cet effet, comme des Raspberry Pi.

Travaux futurs

Une limite de notre mécanisme concerne le choix des signataires accrédités. En effet, nous avons expliqué qu'ils sont choisis parmi les utilisateurs du système, sans donner plus de détails. Comme une charge lourde leur est imposée – vérifier *tous* les témoignages, mettre à jour les réputations de *tous* les fournisseurs de service et les signer –, n'importe quel utilisateur ne peut pas en faire partie. Une heuristique pourrait être de les choisir parmi les fournisseurs les plus actifs du système – c'est-à-dire ceux effectuant le plus de transactions. Cependant, une telle approche pourrait mener à une coalition de ces fournisseurs pour qu'ils restent les plus actifs et qu'ils améliorent leur réputation, ce qui freinerait l'arrivée de nouveaux utilisateurs. Une solution possible pour empêcher de telles attaques serait de réduire la charge des signataires accrédités pour permettre à plus de fournisseurs de remplir ce rôle, et d'opérer un roulement. Un telle diminution est possible en sous-divisant les signataires accrédités en groupes de réputation : chaque ensemble de sous-signataires accrédités s'occupe des fournisseurs ayant une réputation particulière, ce qui diminue le nombre de fournisseurs gérés sans restreindre leur vie privée. Étudier de telles approches est primordial pour améliorer la confiance dans le calcul des scores de réputation.

Une deuxième piste concerne la vie privée des fournisseurs de service. En effet, la propriété proposée impose seulement qu'au moment où un client témoigne, deux fournisseurs de même réputation soient indistinguables ; cette propriété ne garantit rien *après* le témoignage et, en pratique, le client apprend l'identifiant du fournisseur avec qui il a interagi. Pour l'instant, les seuls mécanismes garantissant un anonymat pérenne des fournisseurs de service ne permettent pas aux clients d'émettre des témoignages négatifs [And+08 ; BSS10]. Le problème concerne les scores de réputation : comment s'assurer de leur calcul ? Si les fournisseurs maintiennent eux-mêmes leur réputation, comment s'assurer qu'ils prennent bien tous les témoignages en compte, y compris ceux qui leur sont défavorables ? Ces questions restent ouvertes, et y répondre demandera des outils innovants.

Finalement, un axe d'amélioration général des mécanismes de réputation concerne la compréhension des moteurs de réputation. Nous avons montré qu'il existe de nombreux mécanismes évolués : certains permettent de prendre en compte la confiance dans les témoins pour imiter la transitivité de la réputation « bouche à oreille », la théorie de Dempster-Shafer permet aux témoins de faire part de leur incertitude et les HMM permettent de modéliser les comportements dynamiques des fournisseurs de service. Cependant, l'objectif premier des mécanismes de réputation est d'aider leurs utilisateurs à choisir s'il est raisonnable d'interagir avec un fournisseur. Pour qu'un client puisse faire ce choix, il faut qu'il soit capable de comprendre les scores de réputation des fournisseurs. Les HMM sont très précises, mais permettent-elles de déterminer facilement le risque pris en choisissant d'interagir ? L'*ergonomie* des moteurs de réputation n'a pas encore été étudiée, bien qu'elle soit indispensable pour les utilisateurs finaux.

Publications

[Anc+15] Emmanuelle Anceaume, Yann Busnel, Paul Lajoie-Mazenc et Géraldine Texier. « Reputation for Inter-Domain QoS Routing ». Dans : *IEEE International Symposium on Network Computing and Applications (NCA)*. Cambridge, Massachusetts, USA, septembre 2015.

[Anc+13a] Emmanuelle Anceaume, Gilles Guette, Paul Lajoie-Mazenc, Nicolas Prigent et Valérie Viet Triem Tong. « A Privacy Preserving Distributed Reputation Mechanism ». Dans : *IEEE International Conference on Communications (ICC)*. Budapest, Hungary, juin 2013, pages 1951–1956. doi : 10.1109/ICC.2013.6654809.

[Anc+13b] Emmanuelle Anceaume, Gilles Guette, Paul Lajoie-Mazenc, Thomas Sirvent et Valérie Viet Triem Tong. « Signatures de réputation anonymes ». Dans : *Sécurité des Architectures Réseau et des Systèmes d'Information (SARSSI)*. Poster. Mont-de-Marsan, France, septembre 2013.

[Anc+14] Emmanuelle Anceaume, Gilles Guette, Paul Lajoie-Mazenc, Thomas Sirvent et Valérie Viet Triem Tong. « Extending Signatures of Reputation ». Dans : *Privacy and Identity Management for Emerging Services and Technologies*. Tome 421. Nijmegen, The Netherlands, 2014, pages 165–176. doi : 10.1007/978-3-642-55137-6_13.

[Laj12] Paul Lajoie-Mazenc. « Système de réputation préservant la vie privée ». Dans : *Atelier Protection de la Vie Privée (APVP)*. Groix, France, juin 2012.

[Laj+15a] Paul Lajoie-Mazenc, Emmanuelle Anceaume, Gilles Guette, Thomas Sirvent et Valérie Viet Triem Tong. « Mécanisme de réputation distribué préservant la vie privée avec témoignages négatifs ». Dans : *Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications (AlgoTel)*. Beaune, France, juin 2015. url : https://hal.archives-ouvertes.fr/hal-01148072.

[Laj+15b] Paul Lajoie-Mazenc, Emmanuelle Anceaume, Gilles Guette, Thomas Sirvent et Valérie Viet Triem Tong. « Privacy-Preserving Reputation Mechanism : A Usable Solution Handling Negative Ratings ». Dans : *IFIP WG 11.1 International Conference on Trust Management*. Hamburg, Germany, mai 2015. doi : 10.1007/978-3-319-18491-3_7.

Bibliographie

[AH00] Alfarez Abdul-Rahman et Stephen Hailes. « Supporting Trust in Virtual Communities ». Dans : *Hawaii International Conference on System Sciences (HICSS)*. IEEE Computer Society, janvier 2000. DOI : 10.1109/HICSS.2000.926814.

[Abe+10] Masayuki Abe, Georg Fuchsbauer, Jens Groth, Kristiyan Haralambiev et Miyako Ohkubo. « Structure-Preserving Signatures and Commitments to Group Elements ». Dans : *Advances in Cryptology—CRYPTO*. Sous la direction de Tal Rabin. Santa Barbara, California, USA : Springer Berlin Heidelberg, août 2010, pages 209–236. DOI : 10.1007/978-3-642-14623-7_12.

[Ake70] George A. Akerlof. « The Market for "Lemons" : Quality Uncertainty and the Market Mechanism ». Dans : *The Quarterly Journal of Economics* 84.3 (août 1970), pages 488–500. DOI : 10.2307/1879431.

[Aki+13] Joseph A. Akinyele, Christina Garman, Ian Miers, Matthew W. Pagano, Michael Rushanan, Matthew Green et Aviel D. Rubin. « Charm : a framework for rapidly prototyping cryptosystems ». Dans : *Journal of Cryptographic Engineering* 3.2 (2013), pages 111–128. ISSN : 2190-8508. DOI : 10.1007/s13389-013-0057-3.

[Anc+13a] Emmanuelle Anceaume, Gilles Guette, Paul Lajoie-Mazenc, Nicolas Prigent et Valérie Viet Triem Tong. « A Privacy Preserving Distributed Reputation Mechanism ». Dans : *IEEE International Conference on Communications (ICC)*. Budapest, Hungary, juin 2013, pages 1951–1956. DOI : 10.1109/ICC.2013.6654809.

[Anc+13b] Emmanuelle Anceaume, Gilles Guette, Paul Lajoie-Mazenc, Thomas Sirvent et Valérie Viet Triem Tong. « Signatures de réputation anonymes ». Dans : *Sécurité des Architectures Réseau et des Systèmes d'Information (SARSSI)*. Poster. Mont-de-Marsan, France, septembre 2013.

[Anc+14] Emmanuelle Anceaume, Gilles Guette, Paul Lajoie-Mazenc, Thomas Sirvent et Valérie Viet Triem Tong. « Extending Signatures of Reputation ». Dans : *Privacy and Identity Management for Emerging Services and Technologies*. Tome 421. Nijmegen, The Netherlands, 2014, pages 165–176. DOI : 10.1007/978-3-642-55137-6_13.

[AR06] Emmanuelle Anceaume et Aina Ravoaja. « Incentive-Based Robust Reputation Mechanism for P2P Services ». Dans : *International Conference on Principles of Distributed Systems (OPODIS)*. Sous la direction d'Alexander A. Shvartsman. Bordeaux, France : Springer Berlin Heidelberg, décembre 2006, pages 305–319. DOI : 10.1007/11945529_22.

[And+08] Elli Androulaki, Seung Geol Choi, Steven M. Bellovin et Tal Malkin. « Reputation Systems for Anonymous Networks ». Dans : *Privacy Enhancing Technologies (PETS)*. Sous la direction de Nikita Borisov et Ian Goldberg. Leuven, Belgium : Springer Berlin Heidelberg, juillet 2008, pages 202–218. doi : 10.1007/978-3-540-70630-4_13.

[Ara+11] Diego F. Aranha, Koray Karabina, Patrick Longa, Catherine H. Gebotys et Julio López. « Faster Explicit Formulas for Computing Pairings over Ordinary Curves ». Dans : *Eurocrypt*. Sous la direction de Kenneth G. Paterson. Talinn, Estonia : Springer Berlin Heidelberg, mai 2011, pages 48–68. doi : 10.1007/978-3-642-20465-4_5.

[ASW97] N. Asokan, Matthias Schunter et Michael Waidner. « Optimistic Protocols for Fair Exchange ». Dans : *ACM Conference on Computer and Communications Security (CCS)*. Sous la direction de Richard Graveman, Philippe A. Janson, Clifford Neumann et Li Gong. Zurich, Switzerland : ACM, avril 1997, pages 7–17. doi : 10.1109/49.839935.

[BDK07] Lars Backstrom, Cynthia Dwork et Jon M. Kleinberg. « Wherefore Art Whou R3579X ? Anonymized Social Networks, Hidden Patterns, and Structural Steganography ». Dans : *International Conference on World Wide Web (WWW)*. Sous la direction de Carey L. Williamson, Mary Ellen Zurko, Peter F. Patel-Schneider et Prashant J. Shenoy. Banff, Alberta, Canada : ACM, mai 2007, pages 181–190. doi : 10.1145/1242572.1242598.

[Bal+05] Lucas Ballard, Matthew Green, Breno de Medeiros et Fabian Monrose. *Correlation-Resistant Storage via Keyword-Searchable Encryption*. Cryptology ePrint Archive, Report 2005/417. http://eprint.iacr.org/. 2005.

[Bau+70] Leonard E. Baum, Ted Petrie, George Soules et Norman Weiss. « A Maximization Technique Occurring in the Stastical Analysis of Probabilistic Functions of Markov Chains ». Dans : *The Annals of Mathematical Statistics* 41.1 (février 1970), pages 164–171. doi : 10.1214/aoms/1177697196.

[Bau+01] Roy F. Baumeister, Ellen Bratslavsky, Catrin Finkenauer et Kathleen D. Vohs. « Bad Is Stronger Than Good ». Dans : *Review of General Psychology* 5.4 (2001), pages 323–370. doi : 10.1037/1089-2680.5.4.323.

[Bet11] John D. Bethencourt. « Cryptographic Techniques for Privacy Preserving Identity ». Thèse de doctorat. University of California, Berkeley, 2011.

[BSS10] John Bethencourt, Elaine Shi et Dawn Song. « Signatures of Reputation ». Dans : *Financial Cryptography and Data Security (FC)*. Sous la direction de Radu Sion. Tenerife, Canary Islands : Springer Berlin Heidelberg, janvier 2010, pages 400–407. doi : 10.1007/978-3-642-14577-3_35.

[Bla+10] Olivier Blazy, Georg Fuchsbauer, Malika Izabachène, Amandine Jambert, Hervé Sibert et Damien Vergnaud. « Batch Groth-Sahai ». Dans : *Applied Cryptography and Network Security (ACNS)*. Sous la direction de Jianying Zhou et Moti Yung. Beijing, China : Springer Berlin Heidelberg, juin 2010, pages 218–235. doi : 10.1007/978-3-642-13708-2_14.

[BB08] Dan Boneh et Xavier Boyen. « Short Signatures without Random Oracles and the SDH Assumption in Bilinear Groups ». Dans : *Journal of Cryptology* 21.2 (avril 2008), pages 149–177. doi : 10.1007/s00145-007-9005-7.

[BBS04] Dan Boneh, Xavier Boyen et Hovav Shacham. « Short Group Signatures ». Dans : *Advances in Cryptology—CRYPTO*. Sous la direction de Matthew K. Franklin. Santa Barbara, California, USA : Springer Berlin Heidelberg, août 2004, pages 41–55. doi : 10.1007/978-3-540-28628-8_3.

[BF01] Dan Boneh et Matthew K. Franklin. « Identity-Based Encryption from the Weil Pairing ». Dans : *Advances in Cryptology—CRYPTO*. Sous la direction de Joe Killian. Santa Barbara, California, USA, août 2001, pages 213–229. doi : 10.1007/3-540-44647-8_13.

[Bor06] Nikita Borisov. « Computational Puzzles as Sybil Defenses ». Dans : *IEEE International Conference on Peer-to-Peer Computing (P2P)*. Sous la direction d'Alberto Montresor, Adam Wierzbicki et Nahid Shahmehri. Cambridge, United Kingdom : IEEE Computer Society, octobre 2006, pages 171–176. doi : 10.1109/P2P.2006.10.

[Byg02] Lee A. Bygrave. *Data Protection Law, Approaching Its Rationale, Logic and Limits*. Kluwer Law International, août 2002. isbn : 978-9041198709.

[CHL05] Jan Camenisch, Susan Hohenberger et Anna Lysyanskaya. « Compact E-Cash ». Dans : *Eurocrypt*. Sous la direction de Ronald Cramer. Aarhus, Denmark : Springer Berlin Heidelberg, mai 2005, pages 302–321. doi : 10.1007/11426639_18.

[CL01] Jan Camenisch et Anna Lysyanskaya. « An Efficient System for Non-transferable Anonymous Credentials with Optional Anonymity Revocation ». Dans : *Eurocrypt*. Sous la direction de Birgit Pfitzmann. Innsbruck, Austria : Springer Berlin Heidelberg, mai 2001, pages 93–118. doi : 10.1007/3-540-44987-6_7.

[CS97] Jan Camenisch et Markus Stadler. « Efficient Group Signature Schemes for Large Groups (Extended Abstract) ». Dans : *Advances in Cryptology—CRYPTO*. Sous la direction de Burton S. Kaliski Jr. Santa Barbara, California, USA : Springer Berlin Heidelberg, août 1997, pages 410–424. doi : 10.1007/BFb0052252.

[CH07] Elisabetta Carrara et Giles Hogben. *Reputation-based Systems : a security analysis*. Rapport technique. European Network et Information Security Agency (ENISA), décembre 2007. url : http://www.enisa.europa.eu/publications/archive/reputation-based-systems-a-security-analysis/at_download/fullReport (visité le 25/02/2015).

[CH91] David Chaum et Eugène van Heyst. « Group Signatures ». Dans : *Eurocrypt*. Sous la direction de Donald W. Davies. Brighton, UK : Springer Berlin Heidelberg, avril 1991, pages 257–265. doi : 10.1007/3-540-46416-6_22.

[CSK13] Sebastian Clauss, Stefan Schiffner et Florian Kerschbaum. « *k*-Anonymous Reputation ». Dans : *ACM Symposium on Information, Computer and Communications Security (ASIACCS)*. Sous la direction de Kefei Chen, Qi Xie, Weidong Qiu, Ninghui Li et Wen-Guey Tzeng. ACM, mai 2013, pages 359–368. doi : 10.1145/2484313.2484361.

[Cli+02] Chris Clifton, Murat Kantarcioglu, Jaideep Vaidya, Xiaodong Lin et Michael Y. Zhu. « Tools for Privacy Preserving Distributed Data Mining ». Dans : *SIGKDD Explorations* 4.2 (2002). Sous la direction d'Usama Fayyad, Sunita Sarawagi, Paul Bradley et Johannes Gehrke, pages 28–34. doi : 10.1145/772862.772867.

[Del00] Chrysanthos Dellarocas. « Immunizing Online Reputation Reporting Systems Against Unfair Ratings and Discriminatory Behavior ». Dans : *ACM Conference on Electronic Commerce (EC)*. Sous la direction de Jeffrey K. MacKie Mason et Doug Tygar. Minneapolis, Minnesota, USA : ACM, octobre 2000, pages 150–157. doi : 10.1145/352871.352889.

[Del06] Chrysanthos Dellarocas. « Reputation Mechanisms ». Dans : *Handbook on Economics and Information systems*. Sous la direction de Terrence Hendershott. Tome 1. Elsevier Publishing, 2006, pages 629–660. isbn : 978-0444517715.

[DMS04] Roger Dingledine, Nick Mathewson et Paul F. Syverson. « Tor : The Second-Generation Onion Router ». Dans : *USENIX Security Symposium*. Sous la direction de Matt Blaze. San Diego, California, USA : USENIX, août 2004, pages 303–320. url : http://www.usenix.org/publications/library/proceedings/sec04/tech/dingledine.html.

[Dou02] John R. Douceur. « The Sybil Attack ». Dans : *Peer-to-Peer Systems*. Sous la direction de Peter Druschel, Frans Kaashoek et Antony Rowstron. Cambridge, Massachusets, USA : Springer Berlin Heidelberg, mars 2002, pages 251–260. doi : 10.1007/3-540-45748-8_24.

[DR09] Thai Duong et Juliano Rizz. *Flickr's API Signature Forgery Vulnerability*. Septembre 2009. url : http://netifera.com/research/flickr_api_signature_forgery.pdf (visité le 04/05/2015).

[Elg85] Taher Elgamal. « A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms ». Dans : *IEEE Transactions on Information Theory* 31.4 (juillet 1985), pages 469–472. doi : 10.1109/TIT.1985.1057074.

[ES13] Ehab Elsalamouny et Vladimiro Sassone. « An HMM-based Reputation Model ». Dans : *Advances in Security of Information and Communication Networks (SECNET)*. Sous la direction d'Ali Ismail Awad, Aboul-Ella Hassanien et Kensuke Baba. Cairo, Egypt : Springer Berlin Heidelberg, septembre 2013, pages 111–121. doi : 10.1007/978-3-642-40597-69.

[Est+96] Martin Ester, Hans-Peter Kriegel, Jörg Sander et Xiaowei Xu. « A Density-based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. » Dans : *International Conference on Knowledge Discovery and Data Mining (KDD)*. Sous la direction d'Evangelos Simoudis, Jiawei Han et Usama M. Fayyad. Portland, Oregon, USA : AAAI Press, août 1996, pages 226–231. isbn : 1-57735-004-9.

[Eur95] European Parliament and Council of the European Union. « Directive 95/46/EC ». Dans : *Official Journal of the European Communities* L281 (novembre 1995), pages 31–50.

[Fel87] Paul Feldman. « A Practical Scheme for Non-Interactive Verifiable Secret Sharing ». Dans : *Foundations of Computer Science (FOCS)*. Los Angeles, California, USA : IEEE Computer Society, octobre 1987, pages 427–437. doi : 10.1109/SFCS.1987.4.

[FP08] Georg Fuchsbauer et David Pointcheval. « Anonymous Proxy Signatures ». Dans : *Security and Cryptography for Networks (SCN)*. Sous la direction de Rafail Ostrovsky, Roberto De Prisco et Ivan Visconti. Amalfi, Italy : Springer Berlin Heidelberg, septembre 2008, pages 201–217. doi : 10.1007/978-3-540-85855-3_14.

[GPS06] Steven D. Galbraith, Kenneth G. Paterson et Nigel P. Smart. *Pairings for Cryptographers*. Cryptology ePrint Archive, Report 2006/165. http://eprint.iacr.org/. 2006.

[GPS08] Steven D. Galbraith, Kenneth G. Paterson et Nigel P. Smart. « Pairings for cryptographers ». Dans : *Discrete Applied Mathematics* 156.16 (2008), pages 3113–3121. doi : 10.1016/j.dam.2007.12.010.

[Gam90] Diego Gambetta. « Can We Trust Trust ? » Dans : *Trust : Making and Breaking Cooperative Relations*. Sous la direction de Diego Gambetta. Blackwell Publishers, 1990, pages 213–237. isbn : 978-0631175872.

[GM84] Shafi Goldwasser et Silvio Micali. « Probabilistic Encryption ». Dans : *Journal of Computer and System Sciences* 28.2 (1984), pages 270–299. doi : 10.1016/0022-0000(84)90070-9.

[GK11] Michael T. Goodrich et Florian Kerschbaum. « Privacy-Enhanced Reputation-Feedback Methods to Reduce Feedback Extortion in Online Auctions ». Dans : *ACM Conference on Data and Application Security and Privacy (CODASPY)*. Sous la direction de Ravi S. Sandhu et Elisa Bertino. San Antonio, Texas, USA : ACM, février 2011, pages 273–282. doi : 10.1145/1943513.1943550.

[Gro06] Jens Groth. « Simulation-Sound NIZK Proofs for a Practical Language and Constant-Size Group Signatures ». Dans : *International Conference on the Theory and Application of Cryptology and Information Security—ASIACRYPT*. Sous la direction de Xuejia Lai et Kefei Chen. Springer Berlin Heidelberg, décembre 2006, pages 444–459. doi : 10.1007/11935230_29.

[GS08] Jens Groth et Amit Sahai. « Efficient Non-Interactive Proof Systems for Bilinear Groups ». Dans : *Eurocrypt*. Sous la direction de Nigel P. Smart. Istanbul, Turkey : Springer Berlin Heidelberg, avril 2008, pages 415–432. doi : 10.1007/978-3-540-78967-3_24.

[HMV04] Darrel Hankerson, Alfred J. Menezes et Scott Vanstone. *Guide to Elliptic Curve Cryptography*. Springer New York, 2004. isbn : 978-0-387-95273-4.

[Har11] Kristiyan Haralambiev. « Efficient Cryptographic Primitives for Non-Interactive Zero-Know-ledge Proofs and Applications ». Thèse de doctorat. New York University, mai 2011.

[Has10] Omar Hasan. « Privacy Preserving Reputation Systems for Decentralized Environments ». Thèse de doctorat. INSA Lyon, septembre 2010.

[HBB12] Omar Hasan, Lionel Brunie et Elisa Bertino. « Preserving Privacy of Feedback Providers in Decentralized Reputation Systems ». Dans : *Computers & Security* 31.7 (2012), pages 816–826. doi : 10.1016/j.cose.2011.12.003.

[Has+13] Omar Hasan, Lionel Brunie, Elisa Bertino et Ning Shang. « A Decentralized Privacy Preserving Reputation Protocol for the Malicious Adversarial Model ». Dans : *IEEE Transactions on Information Forensics and Security* 8.6 (2013), pages 949–962. doi : 10.1109/TIFS.2013.2258914.

[HZN09] Kevin J. Hoffman, David Zage et Cristina Nita-Rotaru. « A Survey of Attack and Defense Techniques for Reputation Systems ». Dans : *ACM Computing Surveys* 42.1 (décembre 2009), pages 1–31. doi : 10.1145/1592451.1592452.

[Jøs99] Audun Jøsang. « Trust-based Decision-making for Eletronic Transactions ». Dans : *Nordic Workshop on Secure IT Systems (NORDSEC)*. Sous la direction de Louise Yngström et T. Svensson. Stockholm, Sweden, 1999, pages 496–502.

[JI02] Audun Jøsang et Roslan Ismail. « The Beta Reputation System ». Dans : *Bled Electronic Commerce Conference*. Bled, Slovenia, juin 2002.

[JIB07] Audun Jøsang, Roslan Ismail et Colin Boyd. « A Survey of Trust and Reputation Systems for Online Service Provision ». Dans : *Decision Support Systems* 43.2 (mars 2007). Sous la direction d'Eldon Y. Li et Timon C. Du, pages 618–644. doi : 10.1016/j.dss.2005.05.019.

[Jou00] Antoine Joux. « A One Round Protocol for Tripartite Diffie-Hellman ». Dans : *Algorithmic Number Theory*. Sous la direction de Wieb Bosma. Leiden, The Netherlands, juillet 2000, pages 385–394. doi : 10.1007/10722028_23.

[KSG03] Sepandar D. Kamvar, Mario T. Schlosser et Hector Garcia-Molina. « The EigenTrust Algorithm for Reputation Management in P2P Networks ». Dans : *International World Wide Web Conference (WWW)*. Budapest, Hungary : ACM, mai 2003, pages 640–651. doi : 10.1145/775152.775242.

[KL07] Jonathan Katz et Yehuda Lindell. *Introduction to Modern Cryptography*. Chapman et Hall/CRC Press, 2007. isbn : 978-1-58488-551-1.

[KW03] Jonathan Katz et Nan Wang. « Efficiency Improvements for Signature Schemes with Tight Security Reductions ». Dans : *ACM Conference on Computer and Communications Security (CCS)*. Sous la direction de Sushil Jajodia, Vijayalakshmi Atluri et Trent Jaeger. Washington, Columbia, USA : ACM, octobre 2003, pages 155–164. DOI : 10.1145/948109.948132.

[Ker09] Florian Kerschbaum. « A Verifiable, Centralized, Coercion-Free Reputation System ». Dans : *Workshop on Privacy in the Electronic Society (WPES)*. Sous la direction d'Ehab Al-Shaer et Stefano Paraboschi. Chicago, Illinois, USA : ACM, novembre 2009, pages 61–70. DOI : 10.1145/1655188.1655197.

[Laj12] Paul Lajoie-Mazenc. « Système de réputation préservant la vie privée ». Dans : *Atelier Protection de la Vie Privée (APVP)*. Groix, France, juin 2012.

[Laj+15a] Paul Lajoie-Mazenc, Emmanuelle Anceaume, Gilles Guette, Thomas Sirvent et Valérie Viet Triem Tong. « Mécanisme de réputation distribué préservant la vie privée avec témoignages négatifs ». Dans : *Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications (AlgoTel)*. Beaune, France, juin 2015. URL : https://hal.archives-ouvertes.fr/hal-01148072.

[Laj+15b] Paul Lajoie-Mazenc, Emmanuelle Anceaume, Gilles Guette, Thomas Sirvent et Valérie Viet Triem Tong. « Privacy-Preserving Reputation Mechanism : A Usable Solution Handling Negative Ratings ». Dans : *IFIP WG 11.1 International Conference on Trust Management*. Hamburg, Germany, mai 2015. DOI : 10.1007/978-3-319-18491-3_7.

[LSP82] Leslie Lamport, Robert E. Shostak et Marshall C. Pease. « The Byzantine Generals Problem ». Dans : *ACM Transactions on Programming Languages and System* 4.3 (juillet 1982), pages 382–401. DOI : 10.1145/357172.357176.

[Lit74] Émile Littré. *Dictionnaire de la langue française*. Version électronique créée par François Gannaz http://www.littre.org. Librairie Hachette, 1873–1874.

[Liu+11] Siyuan Liu, Jie Zhang, Chunyan Miao, Yin Leng Theng et Alex C. Kot. « iCLUB : An Integrated Clustering-Based Approach to Improve the Robustness of Reputation Systems ». Dans : *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*. Sous la direction de Liz Sonenberg, Peter Stone, Kagan Tumer et Pinar Yolum. Taipei, Taiwan : IFAAMAS, mai 2011, pages 1151–1152. ISBN : 978-0-9826571-7-1.

[MO04] Tobias Mahler et Thomas Olsen. « Reputation Systems and Data Protection Law ». Dans : *eAdoption and the Knowledge Economy : Issues, Applications, Case Studies*. Sous la direction de Paul Cunningham et Miriam Cunningham. Tome 1. IOS Press, octobre 2004, pages 180–187. ISBN : 1-58603-470-7.

[MUO96] Masahiro Mambo, Keisuke Usuda et Eiji Okamoto. « Proxy Signatures for Delegating Signing Operation ». Dans : *ACM Conference on Computer and Communications Security (CCS)*. Sous la direction de Li Gong et Jacques Stern. New Delhi, India, mars 1996, pages 48–57. DOI : 10.1145/238168.238185.

[MG06] Sergio Marti et Hector Garcia-Molina. « Taxonomy of Trust : Categorizing P2P Reputation Systems ». Dans : *Computer Networks* 50.4 (mars 2006). Sous la direction de Raouf Boutaba et Alan Marshall, pages 472–484. doi : 10.1016/j.comnet.2005.07.011.

[Men09] Alfred J. Menezes. « An Introduction to Pairing-based Cryptography ». Dans : *Recent Trends in Cryptography*. Contemporary Mathematics 477 (2009). Sous la direction d'Ignacio Luengo, pages 47–65.

[Mer89] Ralph C. Merkle. « A Certified Digital Signature ». Dans : *Advances in Cryptology—CRYPTO*. Sous la direction de Gilles Brassard. Santa Barbara, California, USA : Springer New York, août 1989, pages 218–238. doi : 10.1007/0-387-34805-0_21.

[NS08] Arvind Narayanan et Vitaly Shmatikov. « Robust De-anonymization of Large Sparse Datasets ». Dans : *IEEE Symposium on Security and Privacy*. Oakland, California, USA : IEEE Computer Society, mai 2008, pages 111–125. doi : 10.1109/SP.2008.33.

[Nat12] National Institute of Standards and Technology. *FIPS 180-4 : Secure Hash Standard (SHS)*. Mars 2012.

[Nat13] National Institute of Standards and Technology. *FIPS 186-4 : Digital Signatures Standard (DSS)*. Juillet 2013.

[Oka92] Tatsuaki Okamoto. « Provably Secure and Practical Identification Schemes and Corresponding Signature Schemes ». Dans : *Advances in Cryptology—CRYPTO*. Sous la direction d'Ernest F. Brickell. Santa Barbara, California, USA, août 1992, pages 31–53. doi : 10.1007/3-540-48071-4_3.

[Pai99] Pascal Paillier. « Public-Key Cryptosystems Based on Composite Degree Residuosity Classes ». Dans : *Eurocrypt*. Sous la direction de Jacques Stern. Prague, Czech Republic : Springer Berlin Heidelberg, mai 1999, pages 223–238. doi : 10.1007/3-540-48910-X_16.

[PRT04] Elan Pavlov, Jeffrey S. Rosenschein et Zvi Topol. « Supporting Privacy in Decentralized Additive Reputation Systems ». Dans : *International Conference on Trust Management*. Sous la direction de Christian Damsgaard Jensen, Stefan Poslad et Theodosis Dimitrakos. Springer Berlin Heidelberg, avril 2004, pages 108–119. doi : 10.1007/978-3-540-24747-0_9.

[PH10] Andreas Pfitzmann et Marit Hansen. *A terminology for talking about privacy by data minimization : Anonymity, Unlinkability, Undetectability, Unobservability, Pseudonymity, and Identity Management*. v0.34. Août 2010. url : http://dud.inf.tu-dresden.de/literatur/Anon_Terminology_v0.34.pdf (visité le 25/02/2015).

[Rat+01] Sylvia Ratnasamy, Paul Francis, Mark Handley, Richard M. Karp et Scott Shenker. « A Scalable Content-Addressable Network ». Dans : *Conference of the Special Interest Group on Data Communications (SIGCOMM)*. Sous la direction de Rene Cruz et George Varghese. San Diego, California, USA : ACM, août 2001, pages 161–172. doi : 10.1145/383059.383072.

[Rav08] Aina Ravoaja. « Mécanismes et architectures P2P robustes et incitatifs pour la réputation ». Thèse de doctorat. Université de Rennes 1, décembre 2008.

[RA07] Aina Ravoaja et Emmanuelle Anceaume. « STORM : A Secure Overlay for P2P Reputation Management ». Dans : *International Conference on Self-Adaptive and Self-Organizing Systems (SASO)*. Boston, Massachussets, USA : IEEE Computer Society, juillet 2007, pages 247–256. doi : 10.1109/SASO.2007.57.

[Ray02] Michel Raynal. « Consensus in Synchronous Systems : A Concise Guided Tour ». Dans : *Pacific Rim International Symposium on Dependable Computing*. Tsukuba, Japan : IEEE, décembre 2002, pages 221–228. doi : 10.1109/PRDC.2002.1185641.

[RZ02] Paul Resnick et Richard Zeckhauser. « Trust Among Strangers in Internet Transactions : Empirical Analysis of eBay's Reputation System ». Dans : *The Economics of the Internet and E-Commerce* 11 (novembre 2002). Sous la direction de Michael R. Baye, pages 127–157.

[Res+00] Paul Resnick, Richard Zeckhauser, Eric Friedman et Ko Kuwabara. « Reputation Systems ». Dans : *Communications of the ACM* 43.12 (décembre 2000), pages 45–48. doi : 10.1145/355112.355122.

[RSA78] Ron Rivest, Adi Shamir et Leonard Adleman. « A Method for Obtaining Digital Signatures and Public-Key Cryptosystems ». Dans : *Communications of the ACM* 21.2 (février 1978). Sous la direction de Susan L. Graham et Ron Rivest, pages 120–126. doi : 10.1145/359340.359342.

[SPT11] Stefan Schiffner, Andreas Pashalidis et Elmar Tischhauser. « On the limits of privacy in reputation systems ». Dans : *ACM workshop on Privacy in the electronic society (WPES)*. Sous la direction d'Yan Chen et Jaideep Vaidya. Chicago, Illinois, USA : ACM, octobre 2011, pages 33–42. doi : 10.1145/2046556.2046561.

[Sch89] Claus-Peter Schnorr. « Efficient Identification and Signatures for Smart Cards ». Dans : *Advances in Cryptology—CRYPTO*. Sous la direction de Gilles Brassard. Santa Barbara, California, USA : Springer, août 1989, pages 239–252. doi : 10.1007/0-387-34805-0_22.

[Sch91] Claus-Peter Schnorr. « Efficient Signature Generation by Smart Cards ». Dans : *Journal of Cryptology* 4.3 (1991), pages 161–174. doi : 10.1007/BF00196725.

[Sha76] Glenn Shafer. *A Mathematical Theory of Evidence*. Princeton University Press, 1976. isbn : 978-0691100425.

[Sha79] Adi Shamir. « How to Share a Secret ». Dans : *Communications of the ACM* 22.11 (novembre 1979). Sous la direction de Ronald L. Rivest, pages 612–613. doi : 10.1145/359168.359176.

[Sho08] Victor Shoup. *A Computational Introduction to Number Theory and Algebra*. 2e édition. Cambridge University Press, décembre 2008. isbn : 978-0-521-51644-0.

[Sil09] Joseph H. Silverman. *The Arithmetics of Elliptic Curves*. 2e édition. Springer-Verlag, mai 2009. isbn : 978-0-387-09493-9.

[Ste08] Sandra Steinbrecher. « Enhancing Multilateral Security in and by Reputation Systems ». Dans : *The Future of Identity in the Information Society*. Sous la direction de Vashek Matyás, Simone Fischer-Hübner, Daniel Cvrcek et Petr Svenda. Brno, Czech Republich : Springer Berlin Heidelberg, septembre 2008, pages 135–150. doi : 10.1007/978-3-642-03315-5_10.

[Sto+03] Ion Stoica, Robert Morris, David Liben-Nowell, David R. Karger, Marinus Frans Kaashoek, Frank Dabek et Hari Balakrishnan. « Chord : a Scalable Peer-to-Peer Lookup Protocol for Internet Applications ». Dans : *IEEE/ACM Transactions on Networking* 11.1 (février 2003), pages 17–32. doi : 10.1109/TNET.2002.808407.

[Syv94] Paul F. Syverson. « A Taxonomy of Replay Attacks ». Dans : *IEEE Computer Security Foundations Workshop (CSFW)*. Franconia, New Hampshire, USA : IEEE Computer Society, juin 1994, pages 187–191. doi : 10.1109/CSFW.1994.315935.

[SGR97] Paul F. Syverson, David M. Goldschlag et Michael G. Reed. « Anonymous Connections and Onion Routing ». Dans : *Security and Privacy*. IEEE Computer Society. Oakland, California, USA, mai 1997, pages 44–54. doi : 10.1109/SECPRI.1997.601314.

[WJI04] A. Whitby, A. Jøsang et J. Indulska. « Filtering Out Unfair Ratings in Bayesian Reputation Systems ». Dans : *International Workshop on Trust in Agent Societies (TRUST)*. New York, New York, USA, novembre 2004.

[YS02] Bin Yu et Munindar P. Singh. « Distributed Reputation Management for Electronic Commerce ». Dans : *Computational Intelligence* 18.4 (novembre 2002). Sous la direction de Bruce Spencer, pages 535–549. doi : 10.1111/1467-8640.00202.

Glossaire

associabilité Deux entités sont associables si un adversaire est capable de savoir si elles sont liées; par exemple, deux transaction sont associables si un adversaire peut savoir si un même client y est impliqué. 11, 32, 49, 53, 69, 84, 85
autorité centrale Autorité contrôlant l'enregistrement des utilisateurs. 7, 42, 71, 75, 94, 103, 117

blanchiment Attaque consistant, pour un fournisseur, à se créer un nouveau compte lorsque sa réputation est trop faible, permettant alors de la remonter au niveau initial. 47
bourrage d'urne Attaque consistant à témoigner de nombreuses fois sur un fournisseur, visant soit à augmenter soit à diminuer sa réputation. 11, 49, 53, 59, 61, 62, 69, 73, 75, 84, 88, 89, 94, 112, 127

client Utilisateur désirant obtenir un service. 5
confiance Sentiment qui fait qu'on se fie à quelqu'un ou à quelque chose [Lit74]. 36

fournisseur de service Utilisateur proposant un ou des services. 5

homomorphe Un schéma de chiffrement homomorphe permet d'effectuer des opérations sur les chiffrés. 18, 20

interaction Ensemble des communications permettant à un client et à un fournisseur d'effectuer une transaction. 6
invalide Témoignage ou score de réputation refusé par un utilisateur honnête. 7

moteur de réputation manière dont sont calculés les scores de réputation. 6, 9, 11, 38, 47, 78

note Avis d'un client sur le comportement d'un fournisseur. 5, 76
  négative Note représentant un client mécontent. 5
  positive Note représentant un client satisfait. 5

preuve de transaction Attestation de participation à une transaction. 5, 55

score de réputation Résumé du comportement passé d'un fournisseur. 6

témoignage Compte-rendu d'une transaction. 6, 76
témoin Client passé d'un fournisseur. 6
transaction Échange d'un service entre un client et un fournisseur. 6

utilisateur

curieux Utilisateur cherchant à apprendre les informations secrètes des autres utilisateurs mais se comportant honnêtement. 7, 54

honnête Utilisateur se comportant conformément au protocole défini par le mécanisme de réputation. 7, 63, 72

malveillant Utilisateur dont le comportement dévie du protocole défini par le mécanisme de réputation. 7, 54, 63, 75

valide Témoignage ou score de réputation accepté par un utilisateur honnête. 7

Acronymes

CDH Diffie Hellman calculatoire (Computational Diffie Hellman). 16, 17, 26, 28

DDH Diffie Hellman décisionnel (Decisional Diffie Hellman). 17, 20, 30, 89, 125, 126

DHT Table de hachage distribuée (Distributed Hash Table). 42, 44, 50, 72–75, 94, 95

EU-CMA Inforgeabilité existentielle face à des attaques adaptatives par messages choisis (Existential unforgeability against adaptive chosen-message attacks). 27, 30, 32, 126–128, 130–132

HMM Modèle de Markov caché (Hidden Markov Model). 44–46, 144

NIZK Preuve de connaissance non-interactive à divulgation nulle de connaissance (Non-Interactive Zero-Knowledge proof of knowledge). 23, 24, 26, 32, 33, 88, 90, 93, 95, 104, 120, 124, 127, 133, 135, 136

PPT Probabiliste en temps polynomial (Probabilistic Polynomial-Time). 8, 16, 17, 19, 21, 27

SXDH Diffie Hellman symétrique externe (Symmetric eXternal Diffie Hellman). 13, 17, 20, 23, 24, 28, 119, 122, 126, 133