RAPPORT TECHNIQUE PROUVÉ

Synthèse des expérimentations

Auteur : Stéphanie Delaune, LSV, CNRS & INRIA & ENS de Cachan
Francis Klay, France Télécom Division R&D

Date : 1 Mai 2007

Rapport PROUVÉ numéro : 10

Version : 1.0

Loria
CNRS UMR 7503,
Campus Scientifique - BP 239
54506 Vandoeuvre-lès-nancy cedex
www.loria.fr

Laboratoire Spécification Vérification
CNRS UMR 8643, ENS Cachan
61, avenue du président-Wilson
94235 Cachan Cedex, France
www.lsv.ens-cachan.fr

Laboratoire Verimag
CNRS UMR 5104,
Univ. Joseph Fourier, INPG
2 av. de Vignate,
38610 Gières, France
www-verimag.imag.fr

Cril Technology
9/11 rue Jeanne Braconnier
92360 Meudon La Foret Cedex, France
www.cril.fr

France Telecom
Div. Recherche et Développement
38, 40 rue du Général Leclerc
92794 Issy Moulineaux Cedex
www.rd.francetelecom.fr

Résumé : Dans ce document nous présentons une synthèse des deux cas d'étude traités durant le projet. Rappelons qu'il s'agit d'une part d'un protocole de commerce électronique et d'autre part d'un protocole de vote. Pour chacun de ces protocoles, nous analysons les résultats obtenus afin de dégager l'apport des travaux issus du projet et les aspects qui n'ont pas pu être complètement traité. Compte tenu des enseignements tirés, dans la dernière partie nous mettons en perspectives les axes de recherches envisageables pour traiter complètement des protocoles aussi complexes que celui du vote électronique.

Synthèse des expérimentations

Stéphanie Delaune, LSV, CNRS & INRIA & ENS de Cachan
Francis Klay, France Télécom Division R&D

1 Mai 2007

1 Introduction

La communication par des canaux publics comme Internet s'est beaucoup développée ces dernières années. Les transactions utilisant ce médium sont de plus en plus nombreuses : communication client-fournisseur, services audiovisuels (vidéo à la demande), services bancaires, porte-monnaie électronique, protocoles d'enchères, vote électronique, ...

Depuis les années 80, on dispose d'algorithmes cryptographiques suffisamment sûrs, mais même si ces moyens algorithmiques remplissent parfaitement leur spécification, les propriétés sécuritaires ne sont pas pour autant toujours satisfaites. Les communications « sécurisées » sont en effet assurées par des protocoles dits *cryptographiques* qui utilisent ces moyens algorithmiques mais sont constitués de plusieurs messages. Plusieurs exemples célèbres ont montré qu'ils peuvent être attaqués (« man in the middle attack », « replay attack », « dictionary attack », ...) même en présence d'un chiffrement parfait. (Voir l'article de synthèse [9] : presque tous les protocoles d'authentification comportent des failles). Les failles qui permettent de telles attaques sont qualifiées de failles logiques et on s'accorde pour penser que l'étude des failles logiques des protocoles est orthogonale à l'étude des failles du système de cryptographie sous-jacent. Ces failles sont relativement subtiles, difficiles à déceler à la simple vue du texte du protocole.

Ces protocoles interviennent dans des communications électroniques en assurant des propriétés de sécurité, ils ont des caractéristiques spécifiques. On les trouve en très grand nombre, souvent sous la forme de petites variantes d'un protocole connu, les failles de sécurité dans l'un de ces protocoles peuvent avoir des conséquences économiques graves, en particulier à cause de leur déploiement à grande échelle. De plus, d'autres aspects tels que le respect de la vie privée entre en ligne de compte, il est donc crucial de pouvoir vérifier formellement les propriétés de ces protocoles. Cependant ces vérifications sont dures et laborieuses, il en résulte donc des coûts non négligeables, une approche automatique de ce travail est importante.

Dans ce contexte nous avons choisi de travailler sur deux cas d'étude pour valider les travaux réalisés dans le cadre du projet PROUVÉ. Le premier est constitué de deux versions d'un protocole de commerce électronique permettant d'émuler électroniquement une transaction entre le porte-monnaie d'un client et la caisse d'un commerçant. La première version de ce protocole est une version classique basée sur un chiffrement symétrique alors que la seconde repose sur une exponentielle modulaire. La seconde version à été développée à France Télécom par M. Girault et J.C. Paillès [13] afin d'obtenir un système plus ouvert à faible coût. Sur le fond, ce cas d'étude correspond à un problème difficile qui devait rester dans le spectre des outils et techniques développés dans le projet. Le second cas d'étude est un protocole de vote électronique qui a été mis au point par J. Traoré [18], ingénieur de recherche chez France Télécom. Ce protocole est basé sur un mécanisme de signature en aveugle et peut être considéré comme un dérivé du protocole de Fujioka, Okamoto et Ohta [12]. Ce cas d'étude est volontairement complexe tant au niveau de la modélisation des propriétés de sécurité que de la description du protocole lui-même. Son but est de tester les limites des outils et techniques développés dans le projet et de définir de nouvelles perspectives de recherche.

Dans la section 2 nous présentons les résultats obtenus pour le porte-monnaie électronique en abordant la formalisation et la vérification avant de terminer par les retombées de l'expérimentation. La section 3 contient une démarche similaire pour le protocole de vote électronique et la section 4 met en avant un ensemble de perspectives de recherche visant à répondre aux problèmes qui se sont posés lors des études.

2 Protocole du porte-monnaie électronique

L'étude des protocoles cryptographiques a pris une importance considérable avec le développement du commerce électronique. Ce dernier terme recouvre pour commencer les échanges chiffrés d'information qui ont lieu lorsque l'on retire de l'argent dans un distributeur de billets à l'aide d'une carte bancaire, où le distributeur de billets s'engage dans un dialogue avec la carte du client et la banque pour vérifier que l'utilisateur de la carte est honnête, dispose de la somme demandée sur son compte et ne pourra pas récupérer les billets demandés sans que son compte en soit débité. Le commerce électronique recouvre également les paiements par carte à l'aide de terminaux portables où l'utilisateur doit confirmer son identité en entrant son code secret à quatre chiffres, et aussi la monnaie électronique qui a pour but d'émuler électroniquement la monnaie courante. Cette monnaie est, pour des montants peu élevés, plus intéressante qu'une transaction bancaire par carte qui a un coût élevé pour le commerçant.

Toutes ces applications demandent des garanties de sécurité élevées, portant sur des propriétés de secret et d'authenticité, mais aussi de nombreuses autres propriétés, parmi lesquelles, la non-duplication (des factures, dans l'intérêt du client), la non-révocation (des commandes, dans l'intérêt du commerçant), ... Pour assurer ces propriétés de sécurité, des moyens algorithmiques, tels que les chiffrements et les fonctions à sens unique ont été mis au point ; ils permettent d'assurer certaines propriétés, par exemple qu'il est très improbable qu'un individu puisse obtenir un message en clair à partir d'un chiffré sans connaître la clef de déchiffrement.

Dans ce contexte nous avons traité deux versions d'un protocole permettant la réalisation d'une transaction à l'aide d'un porte-monnaie électronique matérialisé par une carte à puce. Bien que les deux protocoles aient pour but la même fonctionnalité, ils sont très différents en raison des primitives cryptographiques utilisées.

Le premier protocole étudié est un porte-monnaie électronique à chiffrement symétrique. Ce protocole permet la réalisation d'une transaction entre un porte-monnaie électronique et un serveur : le but est de garantir un bon niveau de sécurité avec de bonnes performances grâce à l'utilisation du chiffrement symétrique (moins coûteux que les mécanismes de chiffrement asymétrique).

Le second protocole étudié est un porte-monnaie électronique à chiffrement asymétrique développé par France Telecom [13]. Ce protocole permet la réalisation d'une transaction entre un porte-monnaie électronique et un serveur : le but est de garantir un bon niveau de sécurité et de gagner en ouverture par rapport à l'approche symétrique. Ce protocole utilise une méthode de chiffrement asymétrique à faible coût fondée sur une exponentielle modulaire. Le point délicat est que le protocole fait intervenir certaines propriétés algébriques de l'opérateur d'exponentiation. Ceci implique un affaiblissement de l'hypothèse du chiffrement parfait qui est le point particulièrement dur de ce protocole.

Le but de ce premier cas d'étude était de définir les évolutions à apporter aux outils du projet. C'est pourquoi il a été conduit sur ensemble représentatif d'outils : HERMÈS et CASRUL qui sont des outils du projet et PROVERIF [2].

2.1 Formalisation

Comportement. Pour les deux version du protocole, c'est essentiellement la prise en compte du contexte d'exécution des protocoles qui a posé des problèmes dans la plupart des outils. La séquentialité des sessions, issue d'une contrainte physique (à un instant donné, une seule carte peut-être insérée dans le lecteur) n'a pas pu être prise en compte par les différents outils. De ce point de vue, PROVERIF est l'outil le mieux placé, il utilise une algèbre de processus pour décrire le contexte, ce formalisme est flexible, simple et naturel à appréhender. L'utilisation de variables qui perdurent au fil des sessions est un aspect qui n'est pas présent dans tous les outils, or le solde des différents rôles est une variable de ce type.

Pour la formalisation de la version asymétrique il a fallu traiter spécifiquement l'exponentielle modulaire car à l'époque aucun outil n'était capable d'appréhender directement cet opérateur. Afin de prendre en compte les propriétés algébriques de l'exponentielle elle a été codée par chiffrement asymétrique dans la modélisation. Sur le fond, ce codage reste naturelle et, d'un point de vue théorique l'exponentielle peut effectivement être remplacée par un chiffrement asymétrique sans que les caractéristiques du protocole ne soient modifiées. Par contre dans la pratique cette approche n'a pas été retenue par les concepteurs du protocole pour des raisons évidentes de coûts.

Propriétés. Les propriétés a formaliser étaient la *non création de fausse monnaie* et la *non création de faux litiges*.

La *non création de fausse monnaie* passe par un certain équilibre dans les balances des différents agents, mais l'on peut choisir d'énoncer et de vérifier une propriété plus forte en s'intéressant à l'énoncé suivant : « Si le serveur termine une session apparemment avec un certain porte-monnaie P en créditant sa balance d'un montant M alors le porte-monnaie P a bien débité sa balance d'un montant M. ». Le point intéressant est que ce dernier énoncé correspond à une authentification valuée qui est une propriété cryptographique classique présente dans les outils tels que CASRUL ou PROVERIF.

la *non création de faux litiges* correspond à une propriété de non répudiation combinée à une propriété de non duplication. Elle doit assurer qu'en cas de litige le serveur pourra prouver de façon indiscutable sa bonne foi devant un juge en exhibant une information spécifique appelée évidence. En particulier quoi que fasse l'intrus le serveur ne doit jamais stocker à son insu une fausse évidence.

Faute de primitives adaptées, des propriétés telles que la non répudiation et la non duplication n'ont pas été traitées directement. Pour les formaliser la démarche a été la suivante : dans un premier temps un nouveau rôle représentant le juge a été introduit avec un protocole modélisant le rendu de verdict. Ensuite la non création de faux litiges a été codée via une combinaison d'authentifications valuées.

2.2 Résultats et retombées

- Au niveau symbolique et modulo les codages réalisés nous avons pu vérifier pour les deux protocoles la *non création de fausse monnaie* et la *non création de faux litiges* pour un nombre borné de sessions à l'aide de l'outil CASRUL.
- Au niveau symbolique et modulo les codages réalisés nous avons pu vérifier pour les deux protocoles la *non création de fausse monnaie* pour un nombre non borné de sessions à l'aide de l'outil PROVERIF.
- A l'aide de l'outil HERMES nous avons mis en évidence un point intéressant. Si le contexte n'impose pas une exécution séquentielle du protocole et si le serveur ainsi que le porte-monnaie partage un même identifiant alors la propriété de *non création de faux litiges* n'est pas satisfaite pour la version symétrique du protocole. Ceci signifie par exemple qu'il serait dangereux de déployer sans précaution ce protocole sur un serveur WEB.
- Le problème de la déduction de l'intrus est démontré décidable en présence d'un intrus passif dans [4] qui est un article du projet. Il s'agit d'un résultat inportant puisqu'il implique la vérification de tout invariant de trace ou d'état pour un nombre borné de sessions.
- Un mécanisme pour la prise en compte des propriétés algébriques a été introduit dans HERMES et CASRUL.
- Un mécanisme pour une prise en compte naturelle de la non répudiation a été introduit dans CASRUL. Ce mécanisme est basé sur la notion d'ensemble d'évidences.
- Des constructions permettant de modéliser facilement les notions suivantes ont été introduites dans le langage PROUVÉ : propriétés algébriques, canaux sûrs, scénarios d'exécution, conditionnelles, variables persistantes,....
- Des travaux ont été réalisés dans le cadre du projet pour la prise en compte des propriétés algébriques de l'exponentielle lors de la phase de vérification. On peut noter la prise en compte des exposants multiplicatif pour un nombre borné de sessions dans [7] et [8]. Malheureusement ces résultats ne permettent pas encore de traiter la version asymétrique de notre protocole qui repose sur le segment additif de l'exposant. Une autre approche consiste à réduire le problème sur un problème dans une théorie associative et commutative [10], le résultat obtenu à l'avantage d'être particulièrement générique cependant la résolution du problème résultant reste encore un écueil.

3 Protocole de vote électronique

Le but de cette seconde étude était de tester les limites du langage et des outils du projet PROUVÉ. Ainsi le protocole que nous avons choisi d'étudier est volontairement complexe tant au niveau de la modélisation des propriétés de sécurité que de la description du protocole lui-même.

Contrairement à une idée répandue Il est à noter que le vote traditionnel est loin d'être parfait. En effet, un attaquant pourrait forcer un électeur à ne pas aller voter ou faire en sorte que son vote soit considérer comme un vote nul. Il lui suffit pour cela de surveiller l'électeur pendant 24 heures, ou simplement de consulter les registres pour voir si celui-ci a apposé sa signature. L'attaquant peut également remettre à l'électeur un bulletin signé et vérifier que celui-ci se retrouve bien dans l'urne en assistant au dépouillement. Bien sûr, l'électeur pourra profiter de son passage dans l'isoloir pour échanger le bulletin mais si l'attaquant ne retrouve pas son bulletin dans l'urne, il pourrait y avoir des représailles. D'autre part, la vérifiabilité (la possibilité de vérifier que son vote a été comptabilisé) est loin d'être une chose aisée dans le cadre du vote traditionnel. Un électeur ne souhaitant pas faire confiance à une tierce personne doit alors assister à l'élection et au dépouillement.

Le protocole étudié est un protocole de vote qui a été mis au point chez France Télécom par J.Traoré [18]. Il est basé sur le mécanisme de signature en aveugle et peut être considéré comme un dérivé du protocole de protocole de Fujioka, Okamoto, Ohta (FOO92) [12]. Ce dernier nécessite l'intervention de l'électeur à plusieurs reprises. Il n'est donc pas « vote and go ». En effet, l'électeur pour ne pas révéler son vote va simplement envoyer un engagement. Il doit donc lors d'une première phase obtenir la signature de cet engagement auprès d'une autorité et envoyer cet engagement. Ensuite, une fois cette première phase terminée, il doit faire parvenir une donnée permettant d'ouvrir son engagement.

Or, pour être utilisable en pratique, il est important que l'électeur n'ait pas à intervenir plusieurs fois au cours de la procédure de vote. Pour parer à ce défaut, Ohkubo *et al.* [17] ont modifié le protocole FOO92. Ils proposent de ne plus utiliser un schéma d'engagement qui aboutit nécessairement à un protocole de vote en deux phases, mais un schéma de chiffrement classique associé à un réseau de mélangeurs. Une implémentation de ce schéma a été réalisée, il s'agit de VOTOPIA [14].

J. Traoré a mis en évidence une faille (concernant la vérifiabilité) sur VOTOPIA et propose l'utilisation d'un schéma de signature en aveugle à anonymat révocable pour contourner le problème. C'est ce dernier protocole qui est notre cas d'étude.

Cette étude de cas, contrairement aux protocoles que l'on peut trouver dans [9] présente de nombreuses caractéristiques justifiant son étude au sein du projet PROUVÉ. En effet, un protocole de vote, pour être utilisable, doit vérifier de nombreuses propriétés de sécurité dont certaines semblent contradictoires. Pour assurer ces propriétés, il a fallu mettre en place de nouveaux mécanismes de base, plus complexes que les primitives cryptographiques classiques que sont le chiffrement (symétrique / asymétrique) et les fonctions à sens unique. Ainsi, un tel protocole est intéressant aussi bien du point de vue de la modélisation des propriétés algébriques que de la modélisation des propriétés de sécurité. D'autre part, ces protocoles ont un besoin crucial d'être vérifié : la moindre faille pourrait permettre la réalisation d'une fraude à grande échelle.

3.1 Formalisation

Comportement. Comme nous l'avions déjà remarqué lors de notre première étude de cas, le langage PROUVÉ est relativement complet en particulier en ce qui concerne la description des scénarios. Cette souplesse dans la description du scénario nous a permis de coder le mécanisme de phases du protocole de vote en permettant une synchronisation globale entre les différents processus. Ce type de synchronisation est primordial pour la modélisation de protocoles de vote électronique. En effet, pour garantir l'anonymat des votes, il faut par exemple assurer que la procédure de publication ne commence pas avant que les votants aient fini leur procédure de vote.

La formalisation des propriétés algébriques des primitives cryptographiques n'a également pas posé de problème. La théorie équationnelle inclus les équations decrivant le comportement des clefs publiques, du chiffrement et des signatures. Moins classiquement elle inclus également le comportement de la signature en aveugle à anonymat révocable ainsi que le mécanisme de preuve à divulgation nulle de connaissance nécessaire à la vérification des mélangeurs :

– La signature en aveugle est un schéma introduite par D. Chaum [6], qui permet à une entité d'obtenir d'une autre entité la signature d'un message sans que le signataire ne connaisse son contenu. La signature en aveugle à anonymat révocable est une évolution de ce schéma qui permet à l'aide d'une

autorité compétente (appelée juge) de retrouver l'identité du votant fraudeur et le couple (message, signature) en cas de litige.
Pratiquement chaque électeur va ainsi pouvoir obtenir une signature de son vote par une autorité qui vérifiera avant de signer que l'électeur est bien inscrit sur les listes électorales et qu'il n'a pas déjà voté pour cette élection. Commence ensuite la phase de vote proprement dite au cours de laquelle chaque électeur envoie à l'urne son vote signé. Bien entendu, seuls les votes signés par l'autorité seront comptabilisés.

– Les réseaux de mélangeurs ont été introduit par D. Chaum [5]. Un mélangeur est une boîte noire qui simule une permutation aléatoire. Prenant en entrée des données, son but est de cacher la correspondance entre ces données et celles produites en sortie. Lors de l'utilisation de plusieurs mélangeurs en série, on parle de réseaux de mélangeurs. Un réseau de mélangeur permet de réaliser un canal anonyme et c'est ce point qui est à la base de son utilisation dans certains protocoles de vote.

Le principal problème rencontré lors de la modélisation du protocole est lié à la manipulation des listes. Ces dernières sont massivement utilisées dans le protocole en particulier pour garantir la vérifiabilité. Le langage PROUVÉ possède un type List mais très peu de fonctionnalités sont offertes et si l'on souhaite utiliser d'autres fonctionnalités telles que la suppression d'un élément d'une liste ou le parcours d'une liste pour appliquer une opération sur chaque élément alors il faut définir de nouveaux symboles de fonctions et ajouter des équations à la théorie équationnelle. Ceci est possible, mais devient très rapidement extrêmement lourd.

Un autre problème concerne l'absence de boucle while ... do ..., cependant là il s'agit d'un choix délibéré est mûrement réfléchi. Une telle instruction est en effet complexe à traiter lors de la vérification. Pour les besoins de l'expérimentation les boucles ont simplement été supprimées cependant nous reparlerons de ce point dans les perspectives.

Propriétés. La formalisation des propriétés d'un protocoles de vote est particulièrement délicates car ces dernières sont nombreuses et parfois très subtiles avec en particulier :

Pas de Résultat Partiel. Personne ne doit être capable d'obtenir des résultats partiels, la connaissance de ces résultats pourrait influencer les électeurs n'ayant pas encore voté.

Éligibilité - Double Vote. Seules les personnes autorisées à voter le peuvent, et aucun électeur ne doit pouvoir voter deux fois lors d'une même élection. La première propriété est vérifiée si l'intrus ne peut pas obtenir au cours de la première phase du vote la signature ou le certificat lui permettant de continuer le protocole. La deuxième propriété (pas de double vote) assure le fait qu'un électeur ne puisse pas faire en sorte que son vote soit comptabilisé deux fois. Il faut donc que le scrutateur dispose d'un mécanisme lui permettant de rejeter les messages similaires. Mais attention, il ne faudrait pas non plus rejeter des votes valides.

Secret des Votes (Anonymat). Personne ne doit être capable de faire le rapprochement entre un électeur et son vote. Il ne s'agit pas du secret au sens habituel du terme. En effet, supposons qu'il s'agisse d'un simple référendum, les valeurs *oui* et *non* ne sont pas secrètes, mais bien connues de l'agent malhonnête.

Sans Reçu. Aucun électeur ne doit être capable de prouver la manière dont il a voté. Obtenir ou être capable de construire un reçu de son vote, c'est à dire un document prouvant la manière dont on a voté, permettrait l'achat de vote ou la coercition (forcer quelqu'un à voter d'une certaine manière et s'en assurer ensuite).

Vérifiabilité (Individuellement / Universellement). Chaque électeur peut vérifier que son vote a été comptabilisé. Toute personne doit pouvoir se convaincre que tous les votes valides ont été comptabilisés sans avoir été modifiés.

Ces propriétés ne semblent pas facile à exprimer rigoureusement, et certaines d'entre elles paraissent même contradictoires. En effet, chaque électeur doit pouvoir vérifier que son vote a été pris en compte (individuellement vérifiable), et pourtant il ne doit pas pouvoir prouver à un tiers comment il a voté ! Dans le cadre de notre étude nous avons obtenu les niveaux de formalisation qui suivent :

L'absence de résultat partiel repose en particulier sur le secret des votes. Il s'agit d'un invariant qui stipule que si le vote est secret au début du protocole, il le sera aussi au moment qui précède la publication des résultats. Cette propriété a pu être formalisée avec le langage PROUVÉ.

L'éligibilité est une propriété qui peut se voir comme une propriété d'atteignabilité. Elle peut se modéliser partiellement dans le langage PROUVÉ, pour cela il suffit de donner à l'intrus un *vote challenge*, et de voir si ce vote peut se retrouver dans le résultat final.

L'anonymat a pu être formalisé dans un article issu du projet [15]. Le principe est qu'il y a anonymat si l'intrus ne peut pas différencier le processus où deux votants votent v_1 et v_2 du processus où ils inversent leur vote. Cependant cette propriété n'a pas pu être formalisée avec le langage PROUVÉ car il ne prend pas en compte la notion d'équivalence observationnelle.

La propriété de vote sans reçu est relativement complexe à définir. Elle a été fomalisée et étudiée dans un article issu du projet [11]. Cependant en l'état, l'outil PROUVÉ ne permet pas de modéliser ce type de propriété en particulier à cause de l'absence d'équivalence observationnelle dans le langage de propriétés.

La vérifiabilité est également une propriété particulièrement difficile à définir, elle n'a pas été étudiée dans le cadre du projet.

A l'origine du projet PROUVÉ le but était de traiter automatiquement des propriétés d'invariance, il s'agissait d'un choix délibéré issu de l'état de l'art et de la multitude de protocoles qui garantissent de telles propriétés. Le contenu de cette section montre que pour les protocoles les plus modernes et les plus complexes un tel point de vue n'est plus aussi vrai bien que les propriétés d'invariance restent présentes dans ces protocoles.

3.2 Résultats et retombées

Le protocole de vote de notre cas d'étude n'a pas pu être traité d'un seul tenant et dans un cadre uniforme. Ceci n'a rien d'étonnant et c'était prévu dès le départ du projet. Contrairement au premier cas d'étude (le porte-monnaie électronique) qui était un problème adapté aux outils du projet ici le but était très différent : évaluer les carences et les évolutions potentielles des techniques développées au sein du projet sur un des protocoles les plus complexes.

Globalement les résultats sont très positifs. Ils montrent que même sur un protocole complexe les outils de preuve automatique apportent une aide considérable sur certains aspects comme la preuve de secret du vote ou de l'éligibilité. D'un autre côté certaines propriétés de sûreté qui sortent du spectres des outils existants ont pu être prouvées manuellement mais formellement. Ce dernier point est important car ce travail manuel a permis de mieux apprécier les méthodes qui devront être mises en oeuvre demain pour mécaniser les preuves qui ont été réalisées. Concrètement les retombées de ce cas d'étude sont les suivantes :

- Le secret faible du vote à l'issue de la première phase du protocole a été vérifié automatiquement pour un nombre non borné de sessions à l'aide de l'outil HERMÈS du projet.
- La vérification du secret faible ne permet pas d'exclure des attaques par dictionnaire ou une divulgation partielle du secret. Dans un article issu du projet [15] le secret fort et l'éligibilité sont vérifiés pour le protocole de Fujioka, Okamoto, Ohta [12] qui est plus simple. Les preuves ont été réalisées automatiquement avec PROVERIF [2] qui est capable de prendre en compte l'équivalence observationnelle dans certains cas. Cette article contient également une preuve manuelle mais formelle de l'anonymat pour ce protocole.
- Le réseaux de mélangeurs est une pièce importante de notre protocole car c'est l'élément qui est à la base de l'anonymat. Les mélangeurs de notre cas d'étude sont les plus difficiles a prendre en compte car on peut les voir comme des mélangeurs optimistes et à notre connaissance ils n'ont jamais été traité formellement. Un article issu du projet [16] contient un travail préliminaire pour débroussailler le terrain dans ce sens. Il s'agit de l'étude du schéma de vote de Chaum qui est en fait une version simplifiée de notre réseau de mélangeurs. Ce document est intéressant à plusieurs titres : d'un côté il prouve dans le modèle symbolique l'anonymat de vote pour ce schéma, mais en plus il donne

des conditions sur les primitives cryptographiques pour que ce résultat reste valide dans le modèle calculatoire.
- Comme nous l'avons dit plus haut, l'article [15] issu du projet contient un résultat permettant de traiter via une équivalence observationnelle le problème de l'anonymat.
- Finalement dans l'article [11] issu du projet dont il a été question plus haut on trouve une étude et la formalisation de la propriété de vote sans reçu.

4 Perspectives

Globalement le porte-monnaie a permis d'obtenir des résultats qui offre une aide indéniable aux concepteurs de tels protocoles même si des aspects doivent encore être améliorés. A l'inverse et c'était un choix délibéré la complexité du protocole de vote électronique a mis en évidence des carences dans les outils du projet. Les principaux axes de recherche que ces deux études ont mis en évidence sont les suivants :

- Pour pouvoir traiter complètement le porte-monnaie électronique il est encore nécessaire de pouvoir vérifier automatiquement l'exponentielle modulaire. Ce problème est indécidable, cependant la prise en compte du segment des exposants additifs permettrait de faire une vérification de ce protocole pour un nombre borné de sessions. Des travaux tels que ceux présentés [7], [8] ou [10] vont dans ce sens.
- Un autre point concerne l'absence de boucle while ... do ..., cependant là il s'agit d'un choix délibéré est mûrement réfléchi. Une telle instruction est en effet complexe à traiter lors de la vérification. Le problème est qu'outre le vote électronique, de nouveaux protocoles tels que les protocoles de groupe mettent parfois en œuvre des boucles. Pour appréhender cet aspect plusieurs voies sont envisageables : abstraction, dépliage limité ou déduction des invariants de boucle puisque dans la plupart des cas ils sont simples et répétitifs.
- Un grand enseignement de ce projet est que la notion d'équivalence observationnelle est une pièce maîtresse pour appréhender des protocoles aussi complexes que le vote électronique. Intuitivement cette notion signifie que quoi que fasse un intrus il sera incapable de distinguer deux exécutions. Il s'agit d'une propriété de base fondamentale car elle permet de coder la plupart des propriétés de notre protocole. Aujourd'hui le seul outil d'analyse de protocoles cryptographiques capable de prendre en compte cette notion est PROVERIF [2]. Cependant la technique utilisée impose des limitations qui ne permettent par de traiter des propriétés comme l'anonymat dans notre cas.
- Un autre grand enseignement de ce projet concerne l'importance du modèle calculatoire. Ce point est mis en relief dans le travail réalisé autour des réseaux de mélangeurs. Intuitivement et dans un cas extrême le problème est le suivant : dans le modèle symbolique il peut-être impossible d'invalider une propriété de sûreté alors que dans le modèle calculatoire elle peut-être invalidée avec une probabilité de 99%. Cette remarque ne signifie pas que le modèle symbolique soit inutile bien au contraire, c'est dans ce modèle que les preuves ont le plus de chance de pouvoir être automatisées. Ce que signifie cette remarque c'est qu'il n'est pas possible de travailler dans le modèle symbolique en laissant de côté le modèle calculatoire. Quand un cryptologue rédige une preuve manuelle il passe perpétuellement d'un modèle à l'autre selon les aspects qu'il doit traiter. Ceci signifie que les outils a venir devrons être capable de faire de même, pour obtenir ce résultat un principe prometteur a vu le jour en 2000 dans [1]. L'idée est de considérer le modèle symbolique comme une abstraction du modèle calculatoire et de prouver que cette abstraction est sûre. Depuis cet article fondateur ce domaine a donné lieu à une recherche intense dont font partie beaucoup de travaux réalisés dans ce projet.
  Une autre approche potentielle est d'effectuer l'intégralité de la mécanisation du raisonnement formelle dans le modèle calculatoire. C'est cette approche qui a été retenue par Bruno Blanchet dans l'outil CRYPTOVERIF [3]. Actuellement ces deux approches sont en concurrence et il est très difficile d'évaluer quelle voie est la plus prometteuse.

Références

[1] M. Abadi and P. Rogaway. Reconciling two views of cryptography (the computational soundness of formal encryption). In *IFIP International Conference on Theoretical Computer Science (IFIP TCS2000)*, Sendai, Japan, 2000. Springer-Verlag, Berlin Germany.

[2] B. Blanchet. An Efficient Cryptographic Protocol Verifier Based on Prolog Rules. In *Proc. 14th Computer Security Foundations Workshop (CSFW)*, pages 82–96, Cape Breton (Nova Scotia, Canada), 2001. IEEE Comp. Soc. Press.

[3] B. Blanchet. A computationally sound mechanized prover for security protocols. In *IEEE Symposium on Security and Privacy*, pages 140–154, Oakland, California, May 2006.

[4] S. Bursuc, H. Comon-Lundh, and S. Delaune. Deducibility constraints, equational theory and electronic money. In H. Comon-Lundh, C. Kirchner, and H. Kirchner, editors, *Rewriting, Computation and Proof — Essays Dedicated to Jean-Pierre Jouannaud on the Occasion of his 60th Birthday*, volume 4600 of *Lecture Notes in Computer Science*, pages 196–212, Cachan, France, June 2007. Springer.

[5] D. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communication of ACM*, 24(2) :84–88, 1981.

[6] D. Chaum. Blind signature system. In P. Press, editor, *Proc. of CRYPTO '83*, page 153, New York (USA), 1984.

[7] Y. Chevalier, R. Küsters, M. Rusinowitch, and M. Turuani. Deciding the security of protocols with Diffie-Hellman exponentiation and products in exponents. In *FSTTCS*, pages 124–135, 2003.

[8] Y. Chevalier and M. Rusinowitch. Hierarchical combination of intruder theories. In *RTA*, pages 108–122, 2006.

[9] J. Clark and J. Jacob. A survey of authentication protocol literature. 1997.

[10] H. Comon-Lundh and S. Delaune. The finite variant property : How to get rid of some algebraic properties. In J. Giesl, editor, *Proceedings of the 16th International Conference on Rewriting Techniques and Applications (RTA'05)*, volume 3467 of *Lecture Notes in Computer Science*, pages 294–307, Nara, Japan, Apr. 2005. Springer.

[11] S. Delaune, S. Kremer, and M. D. Ryan. Receipt-freeness : Formal definition and fault attacks (extended abstract). In *Proc. Workshop Frontiers in Electronic Elections (FEE'05)*, Milan, Italy, 2005.

[12] A. Fujioka, T. Okamoto, and K. Ohta. A practical secret voting scheme for large scale elections. In J. Seberry and Y. Zheng, editors, *Advances in Cryptology (AUSCRYPT'92)*, volume 718 of *LNCS*, pages 244–251. Springer, 1992.

[13] M. Girault and J. Paillès. Contactless EP : A Public-Key Solution with Good Performances. 2001.

[14] K. Kim, J. Kim, B. Lee, and G. Ahn. Experimental design of worldwide internet voting system using PKI. In *SSGRR'01*, L' Aquila (Italy), 2001.

[15] S. Kremer and M. D. Ryan. Analysis of an electronic voting protocol in the applied pi-calculus. In *Proc. 14th European Symposium on Programming (ESOP'05)*, volume 3444 of *LNCS*, pages 186–200, Edinburgh, U.K., 2005. Springer.

[16] Y. Lakhnech and L. Mazaré. Probabilistic opacity for a passive adversary and its application to chaumś voting scheme. Technical Report 4, Verimag, 2005.

[17] M. Ohkubo, F. Miura, M. Abe, A. Fujioka, and T. Okamoto. An improvement on a practical secret voting scheme. In *Proc. 2nd International Workshop on Information Security (ISW'99)*, volume 1729 of *LNCS*, pages 225–234. Springer, 1999.

[18] J. Traoré. Are blind signatures suitable for on-line voting ? (extended abstract). In *Proc. of Workshop Frontiers in Electronic Elections (FEE'05)*, Milan, Italy, 2005.