Spécificités des protocoles de vote électronique

Stéphanie Delaune Steve Kremer

16 Janvier 2009

Résumé

Le vote est un moyen d'expression qui possède la caractéristique de ne pas divulguer les opinions individuelles. Le but du vote électronique est de se rapprocher du vote réel en permettant de s'exprimer anonymement dans un environnement informatique. Pour être utilisable en pratique, un tel protocole doit satisfaire de nombreuses propriétés de sécurité : la non-divulgation des votes avant la fin du scrutin, l'anonymat, ... Afin de satisfaire ces propriétés complexes, de nouvelles primitives cryptographiques ont été mises en places. En effet, les mécanismes classiques tels que la signature et le chiffrement ne sont pas suffisants pour assurer le bon fonctionnement de ces protocoles de vote.

Dans ce rapport, nous avons identifié les principales spécificités des protocoles de vote électronique.

1 Introduction

Le vote à *bulletins secrets* est un moyen d'expression qui possède la caractéristique de ne pas divulguer les opinions individuelles. Le but du vote électronique est de se rapprocher du vote réel (et pourquoi pas, de faire mieux) en transposant dans un environnement informatique cette manière de s'exprimer anonymement. Suivant que l'on souhaite ou non garder les outils traditionnels que sont les bureaux de vote et les isoloirs, on est amené à considérer deux types de vote électronique : le vote *hors-ligne* et le vote *en-ligne*.

- Le vote *hors-ligne* consiste à conserver les outils du vote traditionnel que sont les bureaux de vote et les isoloirs et à ajouter une « machine à voter ». L'avantage de cette méthode est de permettre une comptabilisation rapide et efficace des bulletins, mais aussi de permettre aux électeurs de voter dans n'importe quel bureau de vote. Les bulletins seront ensuite acheminés par le réseau informatique vers leur véritable destination. Des exemples de tels protocoles sont Prêt à Voter [10, 3], ThreeBallot [12] et Punchscan [11].
- Le vote *en-ligne* permet à un électeur de voter de chez lui en utilisant une simple connexion internet. Des exemples de tels protocoles sont les protocoles de Fujioka et al. [5], de Lee et al. [8] ou encore de Lee et al. [6]. L'inconvénient de cette méthode est le manque de confidentialité : Comment s'assurer qu'un électeur n'a pas voté sous la menace ?

Les protocoles de vote possèdent des caractéristiques spécifiques. Ils font intervenir différentes catégories de participants (autorités électorales, scrutateurs, électeurs), et surtout un nombre d'électeurs non connu *a priori*, mais

faisant parti d'une liste prédéfinie. En fait, un protocole de vote peut être vu comme un assemblage de sous-protocoles servant chacun à la réalisation d'une tâche spécifique. Citons par exemple, le protocole permettant l'initialisation de l'élection, le protocole de vote proprement dit au cours duquel l'électeur s'engage sur une valeur et le protocole effectuant le comptage des bulletins et annonçant les résultats. Certains de ces sous-protocoles sont exécutés un nombre arbitraire de fois au cours d'une même élection, c'est le cas du protocole faisant intervenir l'électeur. Un protocole de vote électronique est en cela très différent des protocoles que l'on considérait jusqu'à présent servant, par exemple, à l'établissement d'une clef de session ou à la réalisation d'une transaction bancaire. Ces derniers ne faisaient intervenir qu'un nombre fixé de participants (en général 2 ou 3).

Les protocoles de vote électronique, contrairement aux protocoles que l'on peut trouver dans [4], présente de nombreuses caractéristiques. Celles-ci justifient en partie la mise au point de techniques particulières pour mener à bien l'étude de ces protocoles. En effet, un protocole de vote, pour être utilisable doit vérifier de nombreuses propriétés de sécurité dont certaines semblent contradictoires. En effet, chaque électeur doit pouvoir vérifier que son vote a été pris en compte et pourtant il ne doit pas pouvoir prouvé à un tiers comment il a voté ! Dans ce rapport nous ne dressons pas la liste de ces propriétés qui seront décrites plus en détail dans le rapport « D1.2 Formalisation des propriétés ». 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, de tels protocoles sont intéressants 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.

2 Spécificités

Il existe de nombreuses solutions de vote électronique et il est difficile de toutes les exposer. Nous allons cependant dresser une liste (la plus exhaustive possible) des différents mécanismes utilisés fréquemment dans ce type de protocoles.

2.1 Chiffrement homomorphique

Pour assurer la non utilisation de certaines données confidentielles, des mécanismes de camouflage sont mis en place : c'est ce que l'on appelle communément le *chiffrement*. Il existe des algorithmes de chiffrement symétriques et asymétriques, déterministes ou probabilistes. Rappelons qu'un schéma de chiffrement consiste en un triplet d'algorithmes comprenant un algorithme de génération de clefs ainsi que deux algorithmes : un de chiffrement et un de déchiffrement. Les différents mécanismes de chiffrement permettent de garantir certaines propriétés. Il est par exemple très improbable qu'un individu obtienne de l'information sur un chiffré s'il ne connaît pas la clef de déchiffrement.

Il est parfois intéressant d'utiliser des schémas de chiffrement disposant de propriétés supplémentaires. C'est le cas du du chiffrement homomorphique. Le chiffrement homomorphique est une propriété de certains cryptosystèmes per-

mettant de chiffrer un certain nombre de données les unes après les autres et de déchiffrer l'ensemble sans compromettre la confidentialité de chacun des éléments initiaux. Ce type de chiffrement vérifie l'égalité suivante :

$$Enc(m_1, k) \times Enc(m_2, k) = Enc(m_1 + m_2, k),$$

Ils trouvent de nombreuses applications, en particulier dans les protocoles de vote électronique. L'idée est de permettre à l'autorité de comptage de comptabiliser les votes au fur et à mesure (utilisation de l'opération de multiplication). Il faut noter que l'autorité de comptage n'a pas pour l'instant connaissance des résultats partiels tant qu'elle ne possède pas la clef k. À ce moment seulement, elle sera capable en un seul déchiffrement d'obtenir le résultat de l'élection.

La solution du vote électronique utilisant un schéma de chiffrement homomorphique nécessite la mise en place d'autres mécanismes tels que la signature et l'utilisation de preuves de connaissance à divulgation nulle de connaissance. Ce dernier mécanisme permet d'assurer par exemple que l'autorité de comptage a bien fait son travail, et que le résultat correspond bien à la somme des votes qui ont été postés.

2.2 Preuve à divulgation nulle de connaissance

La preuve de connaissance à divulgation nulle de connaissance (de l'anglais *zero-knowledge*) consiste à prouver à une personne, lors d'un protocole interactif, sa connaissance d'un secret, sans rien révéler sur celui-ci. La plupart des protocoles à divulgation nulle de connaissance sont des itérations de protocoles en 3 phases (engagement, défi, réponse), mais ce n'est pas nécessairement le cas.

Par exemple, dans l'étude de cas fourni par France Télécom, nous avons besoin de modéliser le fait que le votant réalise le chiffrement de b avec la clef publique du réseau de mélangeurs $\mathsf{pub}(\mathcal{M})$ et il doit fournir une preuve P à divulgation nulle de connaissance accompagnant le message chiffré $\{b\}_{\mathsf{pub}(\mathcal{M})}$ pour prouver qu'il connaît le message en clair correspondant, *i.e.* b. Il est possible de modéliser cet aspect utilisé dans les protocoles de vote à l'aide de la théorie équationnelle suivante :

```
fun proof/3.
fun checkproof/3.
fun ok/0.

equation checkproof(proof(pencrypt(m,pubk,r),m,pubk),
                    pencrypt(m,pubk,r),
                    pubk) = ok.
```

2.3 Signature

La signature est une manière de prouver que la personne qui a produit le document est bien celle qu'elle prétend être. C'est un mécanisme de base, couramment utilisé en cryptographie. Il comprend un algorithme de génération de clefs, un algorithme de signature et un algorithme de vérification. Ce schéma est relativement proche du schéma de chiffrement. Comme certains outils ne définissent pas cette primitive, un codage simple consiste alors à modéliser la signature par un chiffrement asymétrique avec la clef privée de l'agent. Sinon, le schéma est le suivant :

```
fun pk/1.
fun sign/2.
fun checksign/2.
equation checksign(sign(x,y),pk(y)) = x.
```

2.4 Signature en aveugle

Les schémas de signature en aveugle ont été introduits par D. Chaum [2]. Ils permettent à une entité d'obtenir d'une autre la signature d'un message sans que le signataire ne le connaisse. Ainsi, chaque électeur va 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.

Le principe est simple : une personne va faire signer un message à quelqu'un en faisant en sorte que le signataire du message n'apprenne rien sur son contenu. C'est donc un protocole interactif faisant intervenir deux entités. Le schéma est le suivant :

```
fun blind/2.
fun sign/2.
fun unblind/2.
equation unblind(sign(blind(x,y),z),y) = sign(x,z).
equation unblind(blind(x,y),y) = x.
```

Par exemple, le protocole de Fujioka, Okamoto et Ohta [5] est basé sur ce principe. Le protocole fourni par France Télécom est également basé sur ce principe. Plus précisément, un schéma de *signature en aveugle à anonymat révocable* est utilisé afin de pouvoir, à l'aide d'une autorité compétente (appelée juge), retrouver l'identité du votant fraudeur et le couple (message, signature) en cas de litige. En plus du protocole de signature entre le signataire et l'utilisateur, il faut ajouter un protocole, appelé protocole de révocation, entre le signataire et le juge.

2.5 Signature en aveugle à anonymat révocable

Une variante des schémas de signature en aveugle consiste à rendre cet anonymat révocable. Pour un tel schéma, en plus du signataire et de l'utilisateur, une troisième entité peut intervenir, c'est l'autorité (encore appelée juge). Il faut également ajouter un protocole (appelé protocole de révocation) entre le signataire et l'autorité. Il existe deux types de levée d'anonymat, suivant l'information que l'autorité reçoit du signataire :

1. L'autorité reçoit la partie du protocole de signature venant du signataire et donne une information permettant à n'importe qui de retrouver le message et la signature.
2. À l'aide du message et de la signature, l'autorité permet au signataire de retrouver l'utilisateur ou la partie du protocole correspondant à la signature.

Pour modéliser cette primitive, nous proposons le schéma suivant :

```
fun fairblind/2.
fun sign/2.
fun unblind/2.
fun revmsg/2.
fun revsign/2.

equation unblind(sign(fairblind(x,y),z),y) = sign(x,z).
equation revmsg(fairblind(x,y), sign(fairblind(x,y),z)) = x.
equation revsign(fairblind(x,y), sign(fairblind(x,y),z)) = sign(x,z).
```

Les fonctions revmsg et revsign sont ici considérées comme des fonctions privées, *i.e.* non connues de l'intrus. On aurait pu choisir de considérer ces symboles de fonctions comme des symboles de fonctions publiques et ajouter un troisième argument qui aurait été la clé privée du juge.

2.6 Schéma d'engagement

Un schéma d'engagement permet à une personne de mettre en gage une valeur sans la dévoiler : la valeur gagée pourra être définitivement ou temporairement cachée. Un tel schéma comprend deux protocoles :

- un protocole, commit, au cours duquel la personne s'engage sur une valeur,
- un protocole, open, au cours duquel la personne révèle la donnée sur laquelle elle s'est précédemment engagée.

Contrairement aux fonctions de chiffrement et de signature, souvent prédéfinies dans les outils consacrés à la vérification de protocoles cryptographiques, les primitives commit et open d'un schéma d'engagement ne sont en général pas prédéfinies. Le schéma est le suivant :

```
fun commit/2.
fun open/2.
equation open(commit(x,r),r) = x.
```

2.7 Schéma d'engagement avec "trapdoor"

Il s'agit de pouvoir s'engager sur une valeur en ayant la possibilité de changer d'avis lorsque l'on connaît une valeur particulière appelée trapdoor. Ainsi, si l'attaquant demande au votant de voter pour c, celui-ci pourra utiliser un engagement et montrer à l'attaquant que celui-ci s'ouvre sur la valeur c. Grâce au trapdoor, le votant pourra également ouvrir cet engagement sur la valeur de son choix et voter ainsi pour le candidat de son choix. Ce mécanisme est utilisé dans le protocole [9].

```
fun tdcommit/3.
fun open/2.
fun f/4.
equation open(tdcommit(x,r,td),r) = x.
equation tdcommit(x1,r,td) = tdcommit(x2,f(x1,r,td,x2),td)
```

2.8 Rechiffrement

Ce mécanisme est par exemple utilisé dans le protocole de Lee *et al.* [8]. Il utilise comme mécanisme de base le chiffrement asymétrique probabiliste muni

d'un mécanisme de rechiffrement. Ce dernier permet de modifier le nombre aléatoire utilisé au cours du chiffrement. Ainsi, celui-ci est fonction du nombre utilisé au départ et du nombre utilisé au cours du rechiffrement. Dans le cadre d'un protocole de vote, les deux opérations, i.e. de chiffrement et de rechiffrement, seront effectuées par des entités distinctes. Ainsi, personne ne connaîtra ce nombre aléatoire.

```
fun penc/3.
fun pk/1.
fun decrypt/2.
fun rencrypt/2.
fun f/2.

equation decrypt(penc(m,pk(sk),r), sk) = m.
equation rencrypt(penc(m,pk(sk),r1), r2) = penc(m,pk(sk),f(r1,r2)).
```

Dans ce même protocole, il est important d'assurer que le rechiffrement fourni est bien un chiffrement du message de départ. Une possibilité pourrait être d'utiliser une preuve à divulgation nulle de connaissance. Dans ce protocole, un autre mécanisme assez particulier est utilisé : le mécanisme de preuve a vérifieur désigné.

2.9 Preuve à vérifieur désigné (DVP)

Ce mécanisme a pour but de convaincre une tierce personne d'un certain fait. La particularité de ce mécanisme est que cette preuve ne sera convaincante que pour la personne désignée. Elle ne présentera aucune valeur pour les autres. Cette primitive peut se modéliser comme proposé ci-dessous. Noter que la fonction checkdvp fonctionne également en présence d'un faux dvp crée en utilisant la clef privée du verfieur désigné. Ainsi, si le vérifieur désigné n'a jamais révélé sa clef privée, il peut être certain que le dvp qu'il reçoit est un vrai dvp. En revanche, ce dvp ne peut pas être utilisé pour convaincre une autre personne puisqu'il est très facile pour le vérifieur désigné de construire un faux dvp.

```
fun dvp/4.
fun checkdvp/4.
fun ok/0.

equation checkdvp(dvp(x,rencrypt(x,r),r,pk(sk)),x,rencrypt(x,r),pk(sk))= ok.
equation checkdvp(dvp(x,y,z,skv),x,y,pk(skv)) = ok.
```

2.10 Réseaux de mélangeurs

Introduit par D. Chaum [1], un mélangeur est une boîte noire prenant en entrée un nombre quelconque de données et qui a pour but de cacher la correspondance entre ces données et celles produites en sortie. L'utilisation de plusieurs mélangeurs en série, on parle alors de réseaux de mélangeurs, permet d'être sûr du résultat final dès lors qu'un des mélangeurs a réellement brassé les données.

L'utilisation d'un réseau de mélangeurs permet de réaliser un système de vote électronique de manière très simple. Mais, il est alors important de s'assurer

que chacun des mélangeurs a bien fait son travail, *i.e.* que le mélange a bien été effectué, que des valeurs n'ont pas été modifiées, rajoutées ou enlevées. Lorsque chacun des mélangeurs est à même de fournir une preuve que ce travail a bien été effectué, on parle alors de *réseaux de mélangeurs universellement vérifiables*.

Il semble naturel de modéliser un tel mécanisme par un processus qui attendrait un certains nombre de messages en entrée avant de les réémettre en sortie. L'utilisation de l'opérateur "parallèle" permet d'assurer que ces messages sont réémis dans un autre ordre. Cependant, une telle modélisation ne prendrait pas en compte les mélangeurs malhonnêtes.

2.11 Utilisation des phases

Tous les protocoles de vote fonctionnent en plusieurs phases. En effet, on distingue au moins la phase de vote et la phase de comptage. Cette dernière ne pouvant commencer que lorsque la première est terminée. Il est important de pouvoir modéliser ce mécanisme lorsque l'on souhaite étudier les protocoles de vote. En effet, certains comportements ne sont pas possibles (grâce au mécanisme de phases) et il est important d'écarter ce genre de comportement pour assurer certaines propriétés de sécurité [7].

3 Conclusion

Dans ce rapport nous avons identifié les principales spécificités des protocoles de vote électronique. Ces protocoles sont généralement basés sur des primitives cryptographiques présentant des propriétés algébriques essentielles au bon fonctionnement du protocole. De plus les protocoles sont divisés en sous-protocoles avec des points de synchronisation qui pose des nouveaux problèmes de modélisation.

Références

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

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

[3] D. Chaum, P. Ryan, and S. Schneider. A practical voter verifiable election scheme. In *Proc. European Symposium on Research in Computer Security (ESORICS'05)*, volume 3679 of *LNCS*, pages 118–139. Springer, 2005.

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

[5] 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.

[6] A. Juels, D. Catalano, and M. Jakobsson. Coercion-resistant electronic elections. In *Proc. Workshop on Privacy in the Electronic Society (WPES'05)*. ACM Press, 2005.

[7] 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.

[8] B. Lee, C. Boyd, E. Dawson, K. Kim, J. Yang, and S. Yoo. Providing receipt-freeness in mixnet-based voting protocols. In *Proc. Information Security and Cryptology (ICISC'03)*, volume 2971 of *LNCS*, pages 245–258. Springer, 2004.

[9] T. Okamoto. An electronic voting scheme. In *Proc. IFIP World Conference on IT Tools*, pages 21–30, 1996.

[10] Prêt à Voter. `http://www.pretavoter.com/`.

[11] Punchscan. `http://www.punchscan.org/`.

[12] R. L. Rivest. The Threeballot voting system. Unpublished draft.