Preuves à divulgation nulle de connaissance

Cours MPRI niveau 1

Initiation à la cryptologie

3 et 24 avril 2006

Louis Granboulan

Plan du cours

- Introduction au Zero-Knowledge (3 avril)
  1. Preuve de connaissance d'un secret
  2. Formalisme du Zero-Knowledge
  3. Isomorphisme de graphes
  4. Résultat générique : preuve ZK de NP
- Applications (24 avril)
  5. Identification ZK
  6. Identification et signature
  7. Variantes du Zero-Knowledge

Première partie

Introduction au ZK

3 avril 2006

Louis Granboulan

Plan du cours

1. Preuve de connaissance d'un secret
   a. Machines de Turing
   b. Complexité
   c. Prédicats et preuve de connaissance
2. Formalisme du Zero-Knowledge
3. Isomorphismes de graphes
4. Résultat générique : preuve ZK de NP

5. Identification ZK
6. Identification et signature
7. Variantes du Zero-Knowledge

Machines de Turing

- Modèle de calcul
  - Machine de Turing à plusieurs rubans
- Machine probabiliste
  - L'un des rubans contient de l'aléa
- Machine à oracle
  - L'un des rubans sert à interagir avec un oracle calculant en un pas de temps une fonction donnée
- Machine interactive
  - Des rubans d'interaction servent à la communication

Complexité

- Complexité asymptotique
  - Paramètre de sécurité k
    - taille des données polynomiale en k
    - temps de calcul polynomial en k
    - probabilité de succès ε inverse d'un polynôme en k
- Complexité concrète
  - Paramètre de sécurité k
    - temps de calcul t
    - probabilité de succès ε
    - t/ε inférieur à 2^k

Preuve de connaissance

- Prédicat
  - Relation R(x,y) pouvant être calculée
  - Exemple : prédicat polynomial
    - classe NP : sur une donnée x, recherche de y tel que R(x,y)
- Preuve de connaissance
  - Protocole interactif à deux partenaires, ayant en commun un prédicat polynomial R et une valeur x
  - Un partenaire *(prouveur)* connaît y *(témoin, secret)* tel que R(x,y)
  - L'autre partenaire *(vérifieur)* doit être convaincu que le prouveur connaît un y

Plan du cours

1. Preuve de connaissance d'un secret
2. Formalisme du Zero-Knowledge
   a. Complétude
   b. Correction (extraction de témoin)
   c. Simulation (divulgation nulle de connaissance : Zero-Knowledge)
3. Isomorphisme de graphes
4. Résultat générique : preuve ZK de NP

5. Identification ZK
6. Identification et signature
7. Variantes du Zero-Knowledge

Complétude

- Interaction entre un prouveur honnête et un vérifieur honnête
- Le vérifieur accepte toujours
  ou bien refuse avec probabilité négligeable

Correction

- Interaction entre un prouveur malhonnête et un vérifieur honnête
  - Si le vérifieur accepte avec probabilité non négligeable, alors le prouveur connaissait le secret
- Extracteur de connaissance (par réduction)
  - Machine qui interagit avec le prouveur malhonnête, en tant qu'oracle, i.e. en en contrôlant entrées et aléa
  - Calcule un témoin
- En anglais : soundness

Simulation

- Interaction entre un prouveur honnête et un vérifieur malhonnête
  - Aucune information sur le secret n'est divulguée
  - D'où le nom *Zero-Knowledge*
- Simulateur
  - Machine qui simule la conversation entre le prouveur et le vérifieur, de façon indistinguable
  - ZK parfait
  - ZK statistique
  - ZK algorithmique

Plan du cours

1. Preuve de connaissance d'un secret
2. Formalisme du Zero-Knowledge
3. Isomorphisme de graphes
   a. Preuve non-interactive
   b. Preuve interactive
   c. Divulgation nulle de connaissance
4. Résultat générique : preuve ZK de NP

5. Identification ZK
6. Identification et signature
7. Variantes du Zero-Knowledge

Isomorphisme de graphes

- Définition
  - La donnée publique est une paire de graphes à n sommets et d'arêtes E_1 et E_2
  - La connaissance secrète est une permutation π de $\{1…n\}$ telle que $\{u,v\} \in E_1 \Leftrightarrow \{\pi(u),\pi(v)\} \in E_2$
- Preuve non-interactive
  - Le prouveur publie π
  - Vérifie complétude et correction, mais pas divulgation nulle de connaissance
  - La divulgation nulle de connaissance n'est possible qu'avec une preuve interactive

Preuve interactive

- On répète k fois l'échange ci-dessous
  - Le prouveur choisit une permutation ρ de $\{1..n\}$ et calcule H image de E_1 par ρ qu'il envoie au vérifieur

    Witness
  - Le vérifieur choisit i = 1 ou 2 et l'envoie au prouveur

    Challenge
  - Si i=1 le prouveur envoie $\sigma=\rho$, sinon il envoie $\sigma=\rho o\pi$

    Response
  - Le vérifieur vérifie que H est l'image de E_i par σ

- Complétude
  - Si les deux partenaires sont honnêtes,
  - le vérifieur accepte toujours
- Correction
  - Un prouveur malhonnête doit deviner i
  - Sinon on l'exécute deux fois avec le même aléa, et on en extrait π

Divulgation nulle de connaissance

- On répète k fois l'échange ci-dessous
  - Le prouveur choisit une permutation ρ de {1..n} et calcule H image de E_1 par ρ qu'il envoie au vérifieur

    Witness
  - Le vérifieur choisit i = 1 ou 2 et l'envoie au prouveur

    Challenge
  - Si i=1 le prouveur envoie $\sigma=\rho$, sinon il envoie $\sigma=\rho o\pi$

    Response
  - Le vérifieur vérifie que H est l'image de E_i par σ

- Simulation
  - On peut simuler parfaitement la conversation si on connaît i à l'avance
  - Le vérifieur ne peut pas faire la différence entre la conversation qu'il a avec le prouveur et une conversation qu'il peut avoir tout seul, en choisissant i et σ

Plan du cours

1. Preuve de connaissance d'un secret
2. Formalisme du Zero-Knowledge
3. Isomorphisme de graphes
4. Résultat générique : preuve ZK de NP
   a. Fonctions à sens unique et mise en gage
   b. Théorème
   c. Preuve du théorème (3-coloriage)

5. Identification ZK
6. Identification et signature
7. Variantes du Zero-Knowledge

Fonctions à sens unique

- Fonction f : $\{0,1\}^* \rightarrow \{0,1\}^*$
- f est calculable par une machine de Turing polynomiale
  - Pour $x \in \{0,1\}^k$, f(x) se calcule en temps polynomial en k
- Aucune machine de Turing polynomiale ne peut calculer f^{-1} avec probabilité non négligeable

- NB : l'existence de fonctions à sens unique implique P≠NP

Mise en gage de bit

- Fonction h de $\{0,1\}^k \times \{0,1\} \rightarrow \{0,1\}^*$
- Deux propriétés
  - Engagement (binding)
    - publier x puis révéler r tel que x=h(r,b)
    - une seule valeur possible pour b
  - Dissimulation (concealement)
    - publier h(r,b) pour un r aléatoire ne révèle rien sur b
    - dissimulation computationnelle (la recherche exhausative est toujours possible)
- Construction à partir de fonctions à sens unique admise

Théorème

- S'il existe une fonction à sens unique, alors tout prédicat polynomial admet une preuve ZK algorithmique pour le problème NP associé.

- Principe de la preuve
  - On construit une preuve ZK d'un problème NP-complet : tricoloriage de graphe
  - La preuve est algorithmique, car on utilise une mise en gage

Mise en gage de couleur

- Propriétés de h : $\{0,1\}^k \times \{B,R,V\} \rightarrow \{0,1\}^*$
  - Engagement
    - h(r,Y) = h(r',Y') implique Y = Y'
  - Dissimulation
    - Aucun test polynomial probabiliste ne distingue les trois distributions $\mathcal{D}_Y$, images de la fonction r → h(r,Y)
- Construction à partir de fonctions à sens unique
  - admise

Tricoloriage de graphe

- Problème du tricoloriage
  - Soit (V,E) un graphe à k sommets
  - Un tricoloriage est une fonction c : V → {B,R,V} telle que $\{u,v\} \in E \Rightarrow c(u) \neq c(v)$
  - C'est un problème NP-complet
- Prédicat polynomial
  - R(x,y) où x est le graphe et y le coloriage
- Isomorphisme
  - Soit π une permutation des couleurs
  - πoc est aussi un coloriage

Protocole ZK pour le tricoloriage

- Le prouveur
  - choisit une permutation $\pi \in \{0,1\}^k$
  - engendre des valeurs r_t aléatoires, pour tout $t \in V$
  - calcule et publie tous les $e_t = h(r_t, \pi oc(t))$
- Le vérifieur
  - choisit une arête (u,v) et la transmet au prouveur
- Le prouveur
  - révèle r_u, r_v, $c_u = \pi oc(u)$ et $c_v = \pi oc(v)$
- Le vérifieur
  - vérifie $e_u = h(r_u, c_u)$, $e_v = h(r_v, c_v)$ et $c_u \neq c_v$

Complétude

Le prouveur

choisit une permutation $\pi \in \{0,1\}^k$

engendre des valeurs r_t aléatoires, pour tout $t \in V$

calcule et publie tous les $e_t = h(r_t, \pi oc(t))$

Le vérifieur

choisit une arête (u,v) et la transmet au prouveur

Le prouveur

révèle r_u, r_v, $c_u = \pi oc(u)$ et $c_v = \pi oc(v)$

Le vérifieur

vérifie $e_u = h(r_u, c_u)$, $e_v = h(r_v, c_v)$ et $c_u \neq c_v$

- Si les deux partenaires sont honnêtes
  - le vérifieur accepte toujours

Correction

Le prouveur

choisit une permutation $\pi \in \{0,1\}^k$

engendre des valeurs r_t aléatoires, pour tout $t \in V$

calcule et publie tous les $e_t = h(r_t, \pi oc(t))$

Le vérifieur

choisit une arête (u,v) et la transmet au prouveur

Le prouveur

révèle r_u, r_v, $c_u = \pi oc(u)$ et $c_v = \pi oc(v)$

Le vérifieur

vérifie $e_u = h(r_u, c_u)$, $e_v = h(r_v, c_v)$ et $c_u \neq c_v$

- Extracteur de connaissance
  - Interaction avec un prouveur utilisant toujours le même aléa
  - Permet de connaître πoc

Simulation

Le prouveur

choisit une permutation $\pi \in \{0,1\}^k$

engendre des valeurs r_t aléatoires, pour tout $t \in V$

calcule et publie tous les $e_t = h(r_t, \pi oc(t))$

Le vérifieur

choisit une arête (u,v) et la transmet au prouveur

Le prouveur

révèle r_u, r_v, $c_u = \pi oc(u)$ et $c_v = \pi oc(v)$

Le vérifieur

vérifie $e_u = h(r_u, c_u)$, $e_v = h(r_v, c_v)$ et $c_u \neq c_v$

- Simulateur du dialogue
  - Si on connaît à l'avance (u,v)
  - On peut choisir e_u et e_v valides
- ZK algorithmique
  - car reposant sur la difficulté d'inverser h
  - si h(r,Y) contenait de l'information sur Y, alors un simulateur ne peut publier l'ensemble des e_t

Bibliographie

- Texte introductif sur Wikipedia
- Liste de références sur
  http://www.cs.ut.ee/~lipmaa/crypto/link/zeroknowledge/

- J.-J. Quisquater, L. Guillou (Crypto'89)
  How to explain Zero-Knowledge protocols to your children
  Crypto'89
  http://www.dice.ucl.ac.be/crypto/publications/1990/alibaba.pdf
- O. Goldreich, S. Micali, and A. Wigderson (FOCS'86 & JACM)
  How to construct zero-knowledge proof systems for NP
  http://www.wisdom.weizmann.ac.il/~oded/gmw1.html

Exercices

- Trouver des preuves Zero-Knowledge de
  - Non-isomorphisme de graphes
  - SAT

Seconde partie

Applications

24 avril 2006

Louis Granboulan

Plan du cours

1. Preuve de connaissance d'un secret
2. Formalisme du Zero-Knowledge
3. Isomorphisme de graphes
4. Résultat générique : preuve ZK de NP

5. Identification ZK
   a. Racines carrées modulo un entier RSA : Fiat-Shamir
   b. Fiat-Shamir à plusieurs secrets
   c. Guillou-Quisquater
6. Identification et signature
7. Variantes du Zero-Knowledge

Connaissance de racines carrées
Fiat-Shamir

- Données — *paramètre de sécurité k=log(n)*
  - n=pq et $x \in Z_n$; secret y tel que $y^2=x$
- Problème équivalent à la factorisation
  - $x^{(p+1)/4}$ est une racine de x modulo p
  - si z est une racine de $x=y^2$, alors gcd(n,y+z) donne un facteur avec probabilité 1/2
- Protocole de preuve ZK — *répété t tours avec log k=o(t)*
  - Le prouveur choisit $r \in Z_n$ et envoie $c=r^2$
  - Le vérifieur envoie un bit b aléatoire
  - Le prouveur envoie $u=ry^b$
  - Le vérifieur vérifie $u^2=cx^b$

Preuve de correction

- Le prouveur (évent. malhonnête) utilise de l'aléa noté ω
- Le vérifieur utilise de l'aléa noté β
- $T_i(\omega,\beta)=1$ signifie qu'après i tours le vérifieur accepte ; $\varepsilon_i=Pr[T_i=1|T_{i-1}=1]$
- Si $Pr[T_t=0]$ négligeable et si log $k=o(t)$ alors il existe un $i\leq t$ tel que $\varepsilon_i\geq 3/4$.
- On note β_i la variante de β qui change la question du i-ième tour
- On dit que (ω,β) est favorable si $T_i(\omega,\beta)=1$ et $T_i(\omega,\beta_i)=1$
- Si $\varepsilon_i\geq 3/4$ l'ensemble des (ω,β) favorables est non négligeable
- On utilise un (ω,β) favorable pour extraire une racine carrée

Preuve de divulgation nulle

- On simule successivement les tours
- Le simulateur, comme d'habitude, doit anticiper la question b
- Il choisit ensuite $u \in Z_n$ aléatoire et en déduit $c=u^2x^{-b}$
- On vérifie que les distributions de probabilité sont identiques
- Si b est choisi au hasard, quelque soit le vérifieur la simulation est bonne avec probabilité 1/2
- Si la simulation est incorrecte, on réessaie de simuler ce tour
- En moyenne, le simulateur finit en temps 2t

Fiat-Shamir à plusieurs secrets

- Schéma
  - On utilise simultanément λ secrets y_i modulo un même nombre n
  - Le challenge est une suite de λ bits b_i
  - La réponse est $u=r\prod y_i^{b_i}$
- Preuve
  - Il faut $\lambda=O(\log k)$ car la simulation doit anticiper λ secrets
- Conclusion
  - Le nombre de tours est divisé par λ

Comparaison de coûts

Fiat-Shamir répété

- Si n fait k bits et qu'on fait t tours, la communication est 2tk
- Des paramètres de sécurité classiques suggèrent k=1024 et t=20
- Cela fait 5 Ko de communication

Fiat-Shamir à plusieurs secrets

- On a t/λ tours et pour chacun la communication est $2k+\lambda$, ce qui fait un total de $t+2tk/\lambda$
- Avec $\lambda=5$ on arrive à 1 Ko de communication

Guillou-Quisquater

- Preuve de connaissance de racine e-ième
- Protocole : on répète t fois
  - Le prouveur choisit $r \in Z_n$ et envoie $c=r^e$
  - Le vérifieur envoie une valeur b aléatoire inférieure à 2^t
  - Le prouveur envoie $u=ry^b$
  - Le vérifieur vérifie $u^e=cx^b$
- La complétude est évidente
- Si $e>2^t$, on peut prouver la correction
- Si $t=O(\log k)$ on peut prouver la divulgation nulle de connaissance

Plan du cours

1. Preuve de connaissance d'un secret
2. Formalisme du Zero-Knowledge
3. Isomorphisme de graphes
4. Résultat générique : preuve ZK de NP

5. Identification ZK
6. Identification et signature
   a. Méthode générale
   b. Exemple : Schnorr
7. Variantes du Zero-Knowledge

Méthode générale

- Un protocole d'identification
  - Preuve de connaissance d'un secret
  - Itération t fois d'un *tour*, avec le vérifieur qui envoie un challenge binaire à chaque tour
  - Zero-Knowledge si le prouveur ne peut pas deviner quel sera le challenge suivant (phrase à mettre à jour)
- Un protocole de signature
  - Preuve que la signature a été fabriquée par un connaisseur du secret
  - Modèle de sécurité : attaque à message choisi, forge existentielle

Méthode générale

- Les challenges sont fabriqués par le message à signer
  - Le prouveur calcule un haché du message
  - Les bits du haché servent de challenge
  - Modèle de l'oracle aléatoire : la fonction de hachage n'est pas sous le contrôle du prouveur, et donne un résultat aléatoire déterministe
  - En pratique, on hache le témoin (aléatoire) et le message ensemble
- Sécurité prouvée
  - Intuition assez ancienne (dès les débuts du ZK)
  - Preuves rigoureuses plus récentes

Identification de Schnorr

- Données
  - p un premier
  - α et β publics
  - α engendre Z_p^*
- Secret
  - Logarithme de β en base α
  - y tel que $\beta = \alpha^y$
- Prouveur
  - Choisit k et envoie $r = \alpha^k$
- Vérifieur
  - Choisit et envoie e
- Prouveur
  - Calcule et envoie s=ye+k
- Vérifieur
  - Vérifie que $\alpha^s = \beta^e$ r

Signature de Schnorr

- Données
  - p un premier
  - α et β publics
  - α engendre Z_p^*
- Secret
  - Logarithme de β en base α
  - y tel que $\beta = \alpha^y$
- Autres paramètres
  - Message m
  - Fonction de hachage h
  - Signature (r,s)
- Signataire
  - Choisit k et calcule $r=\alpha^k$
  - Calcule e=h(m,r)
  - Calcule s=ye+k
  - La signature est (r,s)
- Vérifieur
  - Vérifie que $\alpha^s = \beta^{h(m,r)}$ r

Plan du cours

1. Preuve de connaissance d'un secret
2. Formalisme du Zero-Knowledge
3. Isomorphismes de graphes
4. Résultat générique : preuve ZK de NP

5. Identification ZK
6. Identification et signature
7. Digressions
   a. Argument Zero-Knowledge
   b. Mises en gage

Argument Zero-Knowledge

- Preuve Zero-Knowledge algorithmique
  - Complétude : parfaite
  - Correction : parfaite
  - Simulation : algorithmique
- Utilise une mise en gage
  - Engagement : parfait
  - Dissimulation : algorithmique

- Argument Zero-Knowledge
  - Complétude : parfaite
  - Correction : algorithmique
  - Simulation : parfaite
- Utilise une mise en gage
  - Engagement : algorithmique
  - Dissimulation : parfaite

Mises en gage
basées sur la résiduosité quadratique

- $h(x,b)=x^2m^b \bmod n$
  - n un entier RSA
  - m public n'est pas un carré
- Engagement
  - parfait
  - sinon on aurait $x_1^2m=x_2^2$
- Dissimulation
  - algorithmique
  - sinon le problème de la résiduosité quadratique est facile

- $h(x,b)=x^2m^b \bmod n$
  - n un entier RSA
  - m public est un carré de racine inconnue
- Engagement
  - algorithmique
  - sinon on pourrait calculer une racine de m
- Dissimulation
  - Parfaite

Mises en gage
basées sur le logarithme discret

- $H(x,b) = \alpha^{f(x,b)} \bmod p$
  - $p \equiv 3 \bmod 4$ un premier public
  - $f(x,b)=x$ ou $p-x$, de telle sorte que le second bit de poids faible soit b
- Engagement
  - parfait
- Dissimulation
  - algorithmique
  - car on peut prouver que savoir trouver le second bit de poids faible suffit à pouvoir calculer le logarithme

- $h(x,b) = \alpha^x \beta^b \bmod p$
  - p un premier
  - α et β publics
  - α engendre Z_p^*
- Engagement
  - algorithmique
  - sinon on pourrait calculer le logarithme de β en base α
- Dissimulation
  - Parfaite

Fin

Exercices

Composition

- Prouver que la propriété de divulgation nulle de connaissance est préservée par composition séquentielle (i.e. répétition du protocole)
- Prouver que la propriété de divulgation nulle de connaissance n'est pas préservée par composition parallèle (i.e. exécution simultanée de plusieurs instances du protocole)
- En déduire des définitions de
  - Concurrent Zero-Knowledge
  - Resettable Zero-Knowledge