Preuves Zero-knowledge

Olivier Sanders (Orange)

Preuves

Soit N un module RSA. Alice veut prouver que y mod N est un carré, i.e. y =x^2

Preuves

Soit N un module RSA. Alice veut prouver que y mod N est un carré, i.e. y =x^2

x

Y =x^2 ?

Solution naïve: transmettre x

Révèle x!

Preuves

Soit pk = g^x une clé publique. Alice veut prouver qu'elle connait x.

x

Pk = g^x ?

Solution naïve: transmettre x

Révèle x!

Preuves

Soit pk = g^x une clé publique. Alice veut prouver qu'elle connait x.

x

Pk = g^x ?

Comment prouver sans révéler d'information?

Système de Preuves

Soit {0,1}*, l'ensemble des chaînes de caractères finies

- Un langage L est un sous-ensemble de {0,1}*

- Une relation R est un sous-ensemble de {0,1}*×{0,1}*

Toute relation R définit implicitement un langage L_R:

$$\{y \in \{0,1\}^* : \text{il existe } w \in \{0,1\}^* \text{ tel que } (y,w) \in R\}$$

w est un témoin de y

Système de Preuves

Soit {0,1}*, l'ensemble des chaînes de caractères finies

- Un langage L est un sous-ensemble de {0,1}*

- Une relation R est un sous-ensemble de {0,1}*×{0,1}*

Toute relation R définit implicitement un langage L_R:

$$\{y \in \{0,1\}^* : \text{il existe } w \in \{0,1\}^* \text{ tel que } (y,w) \in R\}$$

w est un témoin de y

Exemple: L = { x^2 mod N} et R =(x^2,x). A noter que y ∈ L peut avoir plusieurs témoins.

Système de Preuves

Un système de preuves pour une relation R est un protocole interactif entre un prouveur P et un vérificateur V tel que

- Pour tout (y,v) dans R, P prend en entrée (y,w) et V prend en entrée y. A la fin de l'interaction V retourne un bit b.
- Complétude: Pour tout (y,w) dans R, V retourne 0 avec probabilité négligeable.
- Validité: Pour tout $y \in \{0,1\}^*$, si V retourne 1 avec probabilité non négligeable alors il existe w tel que (y,w) dans R.

Système de Preuves

Un système de preuves pour une relation R est un protocole interactif entre un prouveur P et un vérificateur V tel que

- Pour tout (y,v) dans R, P prend en entrée (y,w) et V prend en entrée y. A la fin de l'interaction V retourne un bit b.
- Complétude: Pour tout (y,w) dans R, V retourne 0 avec probabilité négligeable.
- Validité: Pour tout y ∈ {0,1} *, si V retourne 1 avec probabilité non négligeable alors il existe w tel que (y,w) dans R.

Le protocole fonctionne si l'affirmation est vraie

Système de Preuves

Un système de preuves pour une relation R est un protocole interactif entre un prouveur P et un vérificateur V tel que

- Pour tout (y,v) dans R, P prend en entrée (y,w) et V prend en entrée y. A la fin de l'interaction V retourne un bit b.

- Complétude: Pour tout (y,w) dans R, V retourne 0 avec probabilité négligeable.

- Validité: Pour tout y ∈ {0,1} *, si V retourne 1 avec probabilité non négligeable alors il existe w tel que (y,w) dans R.

V ne peut être convaincu si l'affirmation est fausse

Zero-Knowledge

Un système de preuves pour une relation R est dit à divulgation nulle (zero-knowledge) si pour tout vérificateur polynomial V, il existe un simulateur S tel que, pour tout (y,w) dans R, S(y) retourne une chaîne str indistinguable des communications entre P(y,w) et V(y).

Zero-Knowledge

Un système de preuves pour une relation R est dit à divulgation nulle (zero-knowledge) si pour tout vérificateur polynomial V, il existe un simulateur S tel que, pour tout (y,w) dans R, S(y) retourne une chaîne str indistinguable des communications entre P(y,w) et V(y).

Le principe du simulateur est de prouver qu'on peut reproduire les échanges *sans connaître le témoin* w.

-> les échanges ne révèlent alors rien sur w

Exemple

Soit N un module RSA. Alice veut prouver que y mod N est un carré, i.e. y =x^2

r aléatoire

R =r^2

R

c

Z = r * x^c

c bit aléatoire

Z^2= R * y^c ?

Exemple

Soit N un module RSA. Alice veut prouver que y mod N est un carré, i.e. y =x^2

r aléatoire
R =r^2

R

c

Z = r * x^c

c bit aléatoire

Z^2= R * y^c ?

- Complet: Z^2 = (r * x^c)^2 = R * y^c

Exemple

Soit N un module RSA. Alice veut prouver que y mod N est un carré, i.e. y =x^2

- Valide? On montre que si y n'est pas un carré, le protocole échoue avec proba ½.

- En reproduisant le protocole n fois on garantit qu'aucun prouveur P ne peut convaincre V avec une probabilité plus grande que (½)^n

Exemple

Zero-knowledge:

- La trace d'une exécution du protocole est (R,c,Z) tel que Z^2= R * y^c.

- On construit le simulateur S suivant:
  - On choisit Z aléatoirement
  - On choisit c aléatoirement
  - On pose R = Z^2 * y^(-c)

- La trace (R,c,Z) satisfait Z^2= R * y^c.

- R est bien distribué car Z aléatoire

Preuve de Connaissance

- Le système de preuve précédent permet de prouver l'appartenance à un langage, i.e. y $\in$ L.

- Il peut être utile de prouver davantage: la connaissance d'un témoin w pour y, *i.e.* tel que (y,w) $\in$ R.

Exemples: « Je prouve connaissance de x tel que y =x^2 mod N»

« Je prouve connaissance de x tel que y =x^e mod N»

« Je prouve connaissance de x tel que y = g^x »

Preuve de Connaissance

Un système de preuve à divulgation nulle pour une relation R est un système de preuve de connaissance si, pour tout prouveur P accepté avec probabilité non négligeable par un vérificateur V, pour tout $y \in \{0,1\}*$ et w tel que $(y;w) \in R$, il existe un extracteur E(y) capable, notamment en contrôlant l'exécution de P(y;w), de retourner un témoin w' valide pour y.

Preuve de Connaissance

Un système de preuve à divulgation nulle pour une relation R est un système de preuve de connaissance si, pour tout prouveur P accepté avec probabilité non négligeable par un vérificateur V, pour tout $y \in \{0,1\}*$ et w tel que $(y;w) \in R$, il existe un extracteur E(y) capable, notamment en contrôlant l'exécution de P(y;w), de retourner un témoin w' valide pour y.

L'existence de l'extracteur est plus forte que la validité:
Non seulement $(y,w) \in R$ mais on peut récupérer w.

Preuve de Connaissance

Un système de preuve zero-knowledge pour une relation R est un système de preuve de connaissance si, pour tout prouveur P accepté avec probabilité non négligeable par un vérificateur V, pour tout $y \in \{0,1\}*$ et w tel que $(y;w) \in R$, il existe un extracteur E(y) capable, notamment en contrôlant l'exécution de P(y;w), de retourner un témoin w' valide pour y.

La preuve est zero-knowledge… mais on peut en extraire de l'information?

Preuve de Connaissance

Un système de preuve zero-knowledge pour une relation R est un système de preuve de connaissance si, pour tout prouveur P accepté avec probabilité non négligeable par un vérificateur V, pour tout $y \in \{0,1\}*$ et w tel que $(y;w) \in R$, il existe un extracteur E(y) capable, notamment **en contrôlant l'exécution** de P(y;w), de retourner un témoin w' valide pour y.

La preuve est zero-knowledge… mais on peut en extraire de l'information?

-> E a plus de pouvoir qu'un vérificateur V. Il peut par exemple « rembobiner » P vu comme une machine de Turing

Protocole de Schnorr

Soit G un groupe d'ordre p engendré par g. Alice veut prouver qu'elle connait x tel que y = g^x

r aléatoire

R =g^r

R

c

z = r + c* x

c aléatoire

g^z= R * y^c ?

- Complet: g^z = g^r * (g^x)^c = R * y^c

Protocole de Schnorr

Zero-knowledge:

- La trace d'une exécution du protocole est (R,c,z) tel que R aléatoire et g^z= R * y^c

- On construit le simulateur S suivant:
  - On choisit z aléatoirement
  - On choisit c aléatoirement
  - On pose R = g^(z)* y^(-c)

- La trace (R,c,Z) satisfait g^z= R * y^c.

- R est bien distribué car z aléatoire

- c est bien distribué si V génère c conformément au protocole (honest vérifier ZK)

Protocole de Schnorr

Soit G un groupe d'ordre p engendré par g. Alice veut prouver qu'elle connait x tel que y = g^x

R

c

z = r + c* x

E

r aléatoire

R =g^r

c aléatoire

g^z= R * y^c ?

- Comment extraire x? z est parfaitement masqué par r

Protocole de Schnorr

Soit G un groupe d'ordre p engendré par g. Alice veut prouver qu'elle connait x tel que y = g^x

R

c

z = r + c* x

E

r aléatoire

R =g^r

c aléatoire

g^z= R * y^c ?

- E exécute une première fois le protocole pour obtenir (R,c,z)

Protocole de Schnorr

Soit G un groupe d'ordre p engendré par g. Alice veut prouver qu'elle connait x tel que y = g^x

R

E

r aléatoire

R =g^r

- E rembobine ensuite Alice jusqu'à la première étape

Protocole de Schnorr

Soit G un groupe d'ordre p engendré par g. Alice veut prouver qu'elle connait x tel que y = g^x

r aléatoire

R =g^r

R

c'

E

c' aléatoire

- E relance Alice sur un autre challenge c'

Protocole de Schnorr

Soit G un groupe d'ordre p engendré par g. Alice veut prouver qu'elle connait x tel que y = g^x

r aléatoire

R =g^r

R

c'

z' = r + c'* x

E

c' aléatoire

g^z'= R * y^c' ?

- E obtient alors un nouveau triplet (R,c',z') pour le même R (et donc le même r)

Protocole de Schnorr

Extraction:

- E dispose donc de (R,c,z) et (R,c’,z’) tels que

$$g^z = R * y^c \;\&\&\; g^{z'} = R * y^{c'}$$

- Ainsi:

$$g^{(z-z')} = y^{(c-c')}$$

- (z-z’)/(c-c’) est le témoin que l’on cherchait

Protocole de Guillou-Quisquater

Soit N un module RSA et e un nombre premier. Alice veut prouver qu'elle connait s tel que y = s^e

r aléatoire

R =r^e

R

c

Z = r * s^c

c<e aléatoire

Z^e = R * y^c ?

Sigma Protocol

Un Sigma protocol est un protocole de preuve en trois passes

Commitment

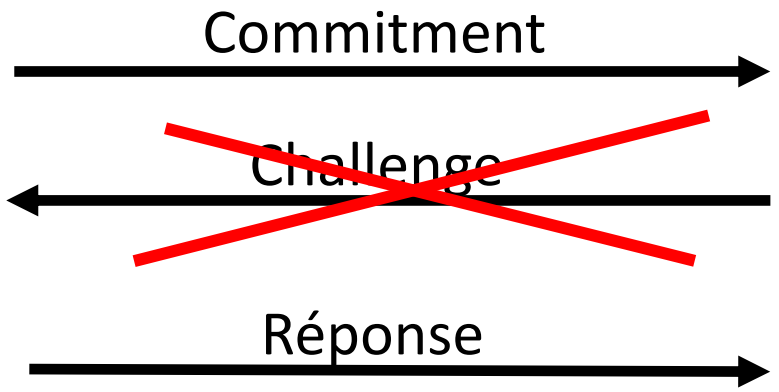
Challenge

Réponse

- Fonctionnement intrinsèquement interactif. La propriété de Zero-knowledge empêche toute vérification de la preuve par un tiers

Fiat-Shamir

Un Sigma protocol est un protocole de preuve en trois passes



Challenge = H(Commitment)

- La technique de Fiat-Shamir permet à Alice de générer ses propres challenges à l'aide d'une fonction de hachage H.

Fiat-Shamir

Un Sigma protocol est un protocole de preuve en trois passes

(Commitment,Réponse)

Challenge = H(Commitment)

- Alice n’a plus besoin d’interactions, toute la preuve se fait en une passe. Preuve vérifiable par un tiers

Schnorr Non-Interactif

Soit G un groupe d'ordre p engendré par g. Alice veut prouver qu'elle connait x tel que y = g^x

r aléatoire
R =g^r
c = H(R)
z = r + c*x

R,z

c = H(R)
g^z= R * y^c ?

- Complet: g^z = g^r * (g^x)^c = R * y^c

Exemple

Zero-knowledge (dans le modèle de l'oracle aléatoire):

- La trace d'une exécution du protocole est (R,z) tel que g^z= R * y^c avec c = H(R)

- On construit le simulateur S suivant:
  - On choisit z aléatoirement
  - On choisit c aléatoirement
  - On pose R = g^z * y^(-c)
  - On programme H(R) = c

- La trace (R,z) satisfait g^z= R * y^c avec c = H(R)

- R est bien distribué car z aléatoire

Schnorr Non-Interactif

Soit G un groupe d'ordre p engendré par g. Alice veut prouver qu'elle connait x tel que y = g^x

r aléatoire
R =g^r
c = H(R)
z = r + c*x

R,z

c = H(R)
g^z= R * y^c ?

- Extractabilité? S exécute une première fois le protocole pour obtenir (R,z)

Schnorr Non-Interactif

Soit G un groupe d'ordre p engendré par g. Alice veut prouver qu'elle connait x tel que y = g^x

r aléatoire
R =g^r
H(R)?

- Extractabilité? S rembobine le prouveur jusqu'à la requête sur R au ROM

Schnorr Non-Interactif

Soit G un groupe d'ordre p engendré par g. Alice veut prouver qu'elle connait x tel que y = g^x

r aléatoire
R =g^r
c' = H(R)
z = r + c'*x

R,z

c' = H(R)
g^z= R * y^c' ?

- Extractabilité? S retourne une autre valeur c' pour H(R)

Schnorr Non-Interactif

Soit G un groupe d'ordre p engendré par g. Alice veut prouver qu'elle connait x tel que y = g^x

r aléatoire
R =g^r
c' = H(R)
z = r + c'*x

R,z

c' = H(R)
g^z= R * y^c' ?

- Extractabilité? A l'aide de (R,z) et (R,z') on retrouve x comme précédemment.

Schnorr Non-Interactif (Alternative)

Soit G un groupe d'ordre p engendré par g. Alice veut prouver qu'elle connait x tel que y = g^x

r aléatoire
R =g^r
c = H(R)
z = r + c*x

c,z

R = g^z * y^(-c)
c = H(R)?

- Complet: g^z * y^(-c) = R
- ZK et Extractabilité se prouvent de la même manière

Schnorr Non-Interactif (Alternative)

Soit G un groupe d'ordre p engendré par g. Alice veut prouver qu'elle connait x tel que $y_1 = g_1^x,\dots,y_n = g_n^x$

r aléatoire

$R_1 = g_1^r$

…

$R_n = g_n^r$

$c = H(R_1,\dots,R_n)$

$z = r + c*x$

c,z

$R_1 = g_1^z * y_1^{(-c)}$

…

$R_n = g_n^z * y_n^{(-c)}$

$c = H(R_1,\dots,R_n)$?

- Intérêt: c plus petit que R et linéarité en le nombre de witness exclusivement

Fiat-Shamir Signature

S=(Commitment,Réponse)

Ou S= (Challenge, Réponse)

Challenge =
H(Commitment,m)

- La méthode Fiat-Shamir permet d'obtenir un système de signature en insérant le message m à signer dans l'entrée de H

Exemple Signature de Schnorr

Soit G un groupe d'ordre p engendré par g. Alice choisit une clé secrète x et une clé publique y = g^x

c,z,m

r aléatoire
R =g^r
c = H(R,m)
z = r + c*x

R = g^z * y^(-c)
c = H(R,m)?

- Vérification de la signature: g^z * y^(-c) = R et c = H(R,m)
- EUF-CMA sûr grâce aux propriétés de ZK et d'extractabilité du protocole sous-jacent