Les canaux subliminaux dans DSA

LSE lightning talk

Marin HANNACHE

10 avril 2018

Plan

1 Introduction rapide à DSA

- Caractéristiques de DSA
- Le problème du logarithme discret
- Signer un message avec DSA
- Remarques sur DSA

2 Canaux subliminaux

- Définition
- Le canal subliminal « haut débit » de DSA
- Les canaux subliminaux « bas débit » de DSA
  - La méthode des résidus quadratiques
  - Les autres méthodes

3 Contre-mesures

- Mise en gage (*commitment scheme*)
- Limites des contre-mesures

Plan

Caractéristiques de DSA

- DSA [1] est un algorithme de signature numérique conçu par David W. KRAVITZ en 1991, alors employé par la NSA. C'est un algorithme de cryptographie asymétrique.
- Il est inspiré d'un autre algorithme de signature mis au point par Taher ELGAMAL en 1984.
- DSA est basé sur le problème du logarithme discret.

1. *Digital signature algorithm*

Caractéristiques de DSA

- DSA [1] est un algorithme de signature numérique conçu par David W. KRAVITZ en 1991, alors employé par la NSA. C'est un algorithme de cryptographie asymétrique.
- Il est inspiré d'un autre algorithme de signature mis au point par Taher ELGAMAL en 1984.
- DSA est basé sur le problème du logarithme discret.

1. *Digital signature algorithm*

Caractéristiques de DSA

- DSA [1] est un algorithme de signature numérique conçu par David W. KRAVITZ en 1991, alors employé par la NSA. C'est un algorithme de cryptographie asymétrique.
- Il est inspiré d'un autre algorithme de signature mis au point par Taher ELGAMAL en 1984.
- DSA est basé sur le problème du logarithme discret.

1. *Digital signature algorithm*

Plan

Le problème du logarithme discret

Il est très simple de calculer une exponentiation modulaire grâce à une variante de l'exponentiation rapide :

Exponentiation modulaire

- $x \equiv 2^{35} \pmod{11}$
- $x = 9$

On ne connaît cependant pas de méthode pour effectuer l'opération inverse en temps polynomial :

Logarithme discret

- $9 \equiv 2^{x} \pmod{11}$
- $x = 35$

Le problème du logarithme discret

Il est très simple de calculer une exponentiation modulaire grâce à une variante de l'exponentiation rapide :

Exponentiation modulaire

- $x \equiv 2^{35} \pmod{11}$
- $x = 9$

On ne connaît cependant pas de méthode pour effectuer l'opération inverse en temps polynomial :

Logarithme discret

- $9 \equiv 2^{x} \pmod{11}$
- $x = 35$

Plan

Signer un message avec DSA

- Paramètres publics :
  1. Une fonction de hachage cryptographique H
  2. Deux nombres premiers p et q tels que $(p-1) \mid q$
  3. Un nombre g dont l'ordre multiplicatif modulo p est q
- Génération d'une paire de clés :
  1. Choisir un nombre x tel que $1 < x < q$
  2. Calculer $y = g^x \mod p$
  3. y est la clé publique, x la clé privée.
- Signer un message m :
  1. On choisit un nombre aléatoire k tel que $1 < k < q$.
  2. On calcule $r = \left(g^k \mod p\right) \mod q$.
  3. On calcule $s = k^{-1}\left(H(m) + rx\right) \mod q$.
  4. La signature est (r, s).

Signer un message avec DSA

- Paramètres publics :
  1. Une fonction de hachage cryptographique H
  2. Deux nombres premiers p et q tels que $(p-1) \mid q$
  3. Un nombre g dont l'ordre multiplicatif modulo p est q
- Génération d'une paire de clés :
  1. Choisir un nombre x tel que $1 < x < q$
  2. Calculer $y = g^x \mod p$
  3. y est la clé publique, x la clé privée.
- Signer un message m :
  1. On choisit un nombre aléatoire k tel que $1 < k < q$.
  2. On calcule $r = \left(g^k \mod p\right) \mod q$.
  3. On calcule $s = k^{-1}\left(H(m) + rx\right) \mod q$.
  4. La signature est (r, s).

Signer un message avec DSA

- Paramètres publics :
  1. Une fonction de hachage cryptographique H
  2. Deux nombres premiers p et q tels que $(p-1) \mid q$
  3. Un nombre g dont l'ordre multiplicatif modulo p est q
- Génération d'une paire de clés :
  1. Choisir un nombre x tel que $1 < x < q$
  2. Calculer $y = g^x \mod p$
  3. y est la clé publique, x la clé privée.
- Signer un message m :
  1. On choisit un nombre aléatoire k tel que $1 < k < q$.
  2. On calcule $r = \left(g^k \mod p\right) \mod q$.
  3. On calcule $s = k^{-1}\left(H(m) + rx\right) \mod q$.
  4. La signature est (r, s).

Signer un message avec DSA

- Paramètres publics :
  1. Une fonction de hachage cryptographique H
  2. Deux nombres premiers p et q tels que $(p-1) \mid q$
  3. Un nombre g dont l'ordre multiplicatif modulo p est q
- Génération d'une paire de clés :
  1. Choisir un nombre x tel que $1 < x < q$
  2. Calculer $y = g^x \mod p$
  3. y est la clé publique, x la clé privée.
- Signer un message m :
  1. On choisit un nombre aléatoire k tel que $1 < k < q$.
  2. On calcule $r = \left(g^k \mod p\right) \mod q$.
  3. On calcule $s = k^{-1}\left(H(m) + rx\right) \mod q$.
  4. La signature est (r, s).

Signer un message avec DSA

- Paramètres publics :
  1. Une fonction de hachage cryptographique H
  2. Deux nombres premiers p et q tels que $(p-1) \mid q$
  3. Un nombre g dont l'ordre multiplicatif modulo p est q
- Génération d'une paire de clés :
  1. Choisir un nombre x tel que $1 < x < q$
  2. Calculer $y = g^x \mod p$
  3. y est la clé publique, x la clé privée.
- Signer un message m :
  1. On choisit un nombre aléatoire k tel que $1 < k < q$.
  2. On calcule $r = \left(g^k \mod p\right) \mod q$.
  3. On calcule $s = k^{-1}\left(H(m) + rx\right) \mod q$.
  4. La signature est (r, s).

Signer un message avec DSA

- Paramètres publics :
  1. Une fonction de hachage cryptographique H
  2. Deux nombres premiers p et q tels que $(p-1) \mid q$
  3. Un nombre g dont l'ordre multiplicatif modulo p est q
- Génération d'une paire de clés :
  1. Choisir un nombre x tel que $1 < x < q$
  2. Calculer $y = g^x \mod p$
  3. y est la clé publique, x la clé privée.
- Signer un message m :
  1. On choisit un nombre aléatoire k tel que $1 < k < q$.
  2. On calcule $r = \left(g^k \mod p\right) \mod q$.
  3. On calcule $s = k^{-1}\left(H(m) + rx\right) \mod q$.
  4. La signature est (r, s).

Plan

Remarques sur DSA

- La dépendance au problème du logarithme discret est immédiate : en sachant le résoudre on peut retrouver la clé privée à partir de la clé publique.
- Le paramètre aléatoire k possède des propriétés intéressantes :
  - La connaissance de k permet de retrouver la clé privée à partir d'une signature.

Démonstration

- *Rappel :* $r = (g^k \mod p) \mod q$
- *Rappel :* $s = k^{-1}(H(m) + rx) \mod q$
- $x = r^{-1}(sk - H(m)) \mod q$

Remarques sur DSA

- La dépendance au problème du logarithme discret est immédiate : en sachant le résoudre on peut retrouver la clé privée à partir de la clé publique.
- Le paramètre aléatoire k possède des propriétés intéressantes :
  - La connaissance de k permet de retrouver la clé privée à partir d'une signature.

Démonstration

- *Rappel :* $r = (g^k \mod p) \mod q$
- *Rappel :* $s = k^{-1}(H(m) + rx) \mod q$
- $x = r^{-1}(sk - H(m)) \mod q$

Remarques sur DSA

- La dépendance au problème du logarithme discret est immédiate : en sachant le résoudre on peut retrouver la clé privée à partir de la clé publique.
- Le paramètre aléatoire k possède des propriétés intéressantes :
  - La connaissance de k permet de retrouver la clé privée à partir d'une signature.

Démonstration

- *Rappel :* $r = (g^k \mod p) \mod q$
- *Rappel :* $s = k^{-1}(H(m) + rx) \mod q$
- $x = r^{-1}(sk - H(m)) \mod q$

Remarques sur DSA

- La dépendance au problème du logarithme discret est immédiate : en sachant le résoudre on peut retrouver la clé privée à partir de la clé publique.
- Le paramètre aléatoire k possède des propriétés intéressantes :
  - La connaissance de k permet de retrouver la clé privée à partir d'une signature.

Démonstration

- *Rappel :* $r = (g^k \mod p) \mod q$
- *Rappel :* $s = k^{-1}(H(m) + rx) \mod q$
- $x = r^{-1}(sk - H(m)) \mod q$

Remarques sur DSA

- La dépendance au problème du logarithme discret est immédiate : en sachant le résoudre on peut retrouver la clé privée à partir de la clé publique.
- Le paramètre aléatoire k possède des propriétés intéressantes :
  - La connaissance de k permet de retrouver la clé privée à partir d'une signature.

Démonstration

- *Rappel :* $r = \left(g^k \mod p\right) \mod q$
- *Rappel :* $s = k^{-1}\left(H(m) + rx\right) \mod q$
- $x = r^{-1}\left(sk - H(m)\right) \mod q$

Remarques sur DSA

- Le paramètre aléatoire k possède des propriétés intéressantes (suite) :
  - La réutilisation du même k pour signer deux messages distincts permet de le dériver des signatures et donc de retrouver la clé privée.

Démonstration

- *Rappel :* $r = (g^k \mod p) \mod q$
- *Rappel :* $s_1 = k^{-1}(H(m_1) + rx) \mod q$
- *Rappel :* $s_2 = k^{-1}(H(m_2) + rx) \mod q$
- $k = (H(m_1) - H(m_2))(s_1 - s_2)^{-1} \mod q$

Remarques sur DSA

- Le paramètre aléatoire k possède des propriétés intéressantes (suite) :
  - La réutilisation du même k pour signer deux messages distincts permet de le dériver des signatures et donc de retrouver la clé privée.

Démonstration

- *Rappel :* $r = (g^k \mod p) \mod q$
- *Rappel :* $s_1 = k^{-1}(H(m_1) + rx) \mod q$
- *Rappel :* $s_2 = k^{-1}(H(m_2) + rx) \mod q$
- $k = (H(m_1) - H(m_2))(s_1 - s_2)^{-1} \mod q$

Remarques sur DSA

- Le paramètre aléatoire k possède des propriétés intéressantes (suite) :
  - La réutilisation du même k pour signer deux messages distincts permet de le dériver des signatures et donc de retrouver la clé privée.

Démonstration

- *Rappel :* $r = \left(g^k \mod p\right) \mod q$
- *Rappel :* $s_1 = k^{-1}\left(H(m_1) + rx\right) \mod q$
- *Rappel :* $s_2 = k^{-1}\left(H(m_2) + rx\right) \mod q$
- $k = \left(H(m_1) - H(m_2)\right)\left(s_1 - s_2\right)^{-1} \mod q$

Remarques sur DSA

- Le paramètre aléatoire k possède des propriétés intéressantes (suite) :
  - La réutilisation du même k pour signer deux messages distincts permet de le dériver des signatures et donc de retrouver la clé privée.

Démonstration

- *Rappel :* $r = \left(g^k \mod p\right) \mod q$
- *Rappel :* $s_1 = k^{-1}\left(H(m_1) + rx\right) \mod q$
- *Rappel :* $s_2 = k^{-1}\left(H(m_2) + rx\right) \mod q$
- $k = \left(H(m_1) - H(m_2)\right)\left(s_1 - s_2\right)^{-1} \mod q$

Remarques sur DSA

- Le paramètre aléatoire k possède des propriétés intéressantes (suite et fin) :
  - La connaissance de la clé privée permet de retrouver k à partir de la signature !

Démonstration

- *Rappel :* $r = (g^k \mod p) \mod q$
- *Rappel :* $s = k^{-1}(H(m) + rx) \mod q$
- $k = s^{-1}(H(m) + rx) \mod q$

Remarques sur DSA

- Le paramètre aléatoire k possède des propriétés intéressantes (suite et fin) :
  - La connaissance de la clé privée permet de retrouver k à partir de la signature !

Démonstration

- *Rappel :* $r = (g^k \mod p) \mod q$
- *Rappel :* $s = k^{-1}(H(m) + rx) \mod q$
- $k = s^{-1}(H(m) + rx) \mod q$

Remarques sur DSA

- Le paramètre aléatoire k possède des propriétés intéressantes (suite et fin) :
  - La connaissance de la clé privée permet de retrouver k à partir de la signature !

Démonstration

- *Rappel :* $r = \left(g^k \mod p\right) \mod q$
- *Rappel :* $s = k^{-1}\left(H(m) + rx\right) \mod q$
- $k = s^{-1}\left(H(m) + rx\right) \mod q$

Remarques sur DSA

- Le paramètre aléatoire k possède des propriétés intéressantes (suite et fin) :
  - La connaissance de la clé privée permet de retrouver k à partir de la signature !

Démonstration

- *Rappel :* $r = \left(g^k \mod p\right) \mod q$
- *Rappel :* $s = k^{-1}\left(H(m) + rx\right) \mod q$
- $k = s^{-1}\left(H(m) + rx\right) \mod q$

Plan

Définition

- Le concept de canal subliminal a été théorisé en 1984 par Gustavus SIMMONS.
- Il s'agit d'un canal de communication sécurisé et caché au sein d'un autre canal de communication qui ne l'est pas.
- Il ne doit pas être possible de distinguer les communications utilisant ces canaux de celles qui ne les utilisent pas.

Définition

- Le concept de canal subliminal a été théorisé en 1984 par Gustavus SIMMONS.
- Il s'agit d'un canal de communication sécurisé et caché au sein d'un autre canal de communication qui ne l'est pas.
- Il ne doit pas être possible de distinguer les communications utilisant ces canaux de celles qui ne les utilisent pas.

Définition

- Le concept de canal subliminal a été théorisé en 1984 par Gustavus SIMMONS.
- Il s'agit d'un canal de communication sécurisé et caché au sein d'un autre canal de communication qui ne l'est pas.
- Il ne doit pas être possible de distinguer les communications utilisant ces canaux de celles qui ne les utilisent pas.

Exemples

RSA, ElGamal et d'autres cryptosystèmes sont dotés de canaux subliminaux

DSA possède un canal « haut débit » de 160 bits et plusieurs canaux « bas débit ». Certains canaux « bas débit » peuvent être utilisés simultanément.

Exemples

RSA, ElGamal et d'autres cryptosystèmes sont dotés de canaux subliminaux

DSA possède un canal « haut débit » de 160 bits et plusieurs canaux « bas débit ». Certains canaux « bas débit » peuvent être utilisés simultanément.

Plan

Le canal « haut débit » de DSA

1. La clé privée du signataire est connue du destinataire.
2. Le message subliminal est chiffré à l'aide d'une méthode convenue à l'avance.
3. Ou lieu de choisir k aléatoirement, on utilise le message chiffré à la place.
4. Le destinataire connaissant la clé privée utilisée, il peut retrouver la valeur de k à partir de la signature et en déduire le message subliminal.

Le canal « haut débit » de DSA

1. La clé privée du signataire est connue du destinataire.
2. Le message subliminal est chiffré à l'aide d'une méthode convenue à l'avance.
3. Ou lieu de choisir k aléatoirement, on utilise le message chiffré à la place.
4. Le destinataire connaissant la clé privée utilisée, il peut retrouver la valeur de k à partir de la signature et en déduire le message subliminal.

Le canal « haut débit » de DSA

1. La clé privée du signataire est connue du destinataire.
2. Le message subliminal est chiffré à l'aide d'une méthode convenue à l'avance.
3. Ou lieu de choisir k aléatoirement, on utilise le message chiffré à la place.
4. Le destinataire connaissant la clé privée utilisée, il peut retrouver la valeur de k à partir de la signature et en déduire le message subliminal.

Le canal « haut débit » de DSA

1. La clé privée du signataire est connue du destinataire.
2. Le message subliminal est chiffré à l'aide d'une méthode convenue à l'avance.
3. Ou lieu de choisir k aléatoirement, on utilise le message chiffré à la place.
4. Le destinataire connaissant la clé privée utilisée, il peut retrouver la valeur de k à partir de la signature et en déduire le message subliminal.

Plan

Plan

Résidus quadratiques

Définition

On dit d'un entier naturel a qu'il est un résidu quadratique modulo n s'il existe un entier x tel que :

$$x^2 \equiv a \pmod{n}$$

On peut faire le vérification rapidement si n est un nombre premier grâce au critère d'Euler :

Critère d'Euler

- Si $a^{(n-1)/2} \equiv 1 \pmod{n}$ alors a est un résidu quadratique modulo n.
- Si $a^{(n-1)/2} \equiv -1 \pmod{n}$ alors a n'est pas un résidu quadratique modulo n.

Résidus quadratiques

Définition

On dit d'un entier naturel a qu'il est un résidu quadratique modulo n s'il existe un entier x tel que :

$$x^2 \equiv a \pmod{n}$$

On peut faire le vérification rapidement si n est un nombre premier grâce au critère d'Euler :

Critère d'Euler

- Si $a^{(n-1)/2} \equiv 1 \pmod{n}$ alors a est un résidu quadratique modulo n.
- Si $a^{(n-1)/2} \equiv -1 \pmod{n}$ alors a n'est pas un résidu quadratique modulo n.

La méthode des résidus quadratiques

1. Les participant conviennent au préalable d'un nombre premier secret P.
2. Le signataire choisi k de façon à ce que r soit un residu quadratique modulo P, s'il veut transmettre un 1 ; ou que r ne le soit pas s'il veut transmettre un 0.
3. Le destinataire applique le critère d'Euler sur r pour retrouver le bit transmi par le canal subliminal.

Il est possible d'échanger plusieurs bits de cette façon en convenant de plusieurs nombre premiers. Mais la probabilité de trouver un k qui est ou n'est pas un résidu quadratique modulo i nombres premiers est de 2^{-i}.

La méthode des résidus quadratiques

1. Les participant conviennent au préalable d'un nombre premier secret P.
2. Le signataire choisi k de façon à ce que r soit un residu quadratique modulo P, s'il veut transmettre un 1 ; ou que r ne le soit pas s'il veut transmettre un 0.
3. Le destinataire applique le critère d'Euler sur r pour retrouver le bit transmi par le canal subliminal.

Il est possible d'échanger plusieurs bits de cette façon en convenant de plusieurs nombre premiers. Mais la probabilité de trouver un k qui est ou n'est pas un résidu quadratique modulo i nombres premiers est de 2^{-i}.

La méthode des résidus quadratiques

1. Les participant conviennent au préalable d'un nombre premier secret P.
2. Le signataire choisi k de façon à ce que r soit un residu quadratique modulo P, s'il veut transmettre un 1 ; ou que r ne le soit pas s'il veut transmettre un 0.
3. Le destinataire applique le critère d'Euler sur r pour retrouver le bit transmi par le canal subliminal.

Il est possible d'échanger plusieurs bits de cette façon en convenant de plusieurs nombre premiers. Mais la probabilité de trouver un k qui est ou n'est pas un résidu quadratique modulo i nombres premiers est de 2^{-i}.

La méthode des résidus quadratiques

1. Les participant conviennent au préalable d'un nombre premier secret P.
2. Le signataire choisi k de façon à ce que r soit un residu quadratique modulo P, s'il veut transmettre un 1 ; ou que r ne le soit pas s'il veut transmettre un 0.
3. Le destinataire applique le critère d'Euler sur r pour retrouver le bit transmi par le canal subliminal.

Il est possible d'échanger plusieurs bits de cette façon en convenant de plusieurs nombre premiers. Mais la probabilité de trouver un k qui est ou n'est pas un résidu quadratique modulo i nombres premiers est de 2^{-i}.

Plan

Les autres méthodes

- Un autre canal subliminal de 1 bit existe, il est basé sur une variante du masque jetable.
- Un canal subliminal de 159 bits existe, il ne nécessite pas de compromettre sa clé privée mais demande plus de calculs de la part du destinataire.

Les autres méthodes

- Un autre canal subliminal de 1 bit existe, il est basé sur une variante du masque jetable.
- Un canal subliminal de 159 bits existe, il ne nécessite pas de compromettre sa clé privée mais demande plus de calculs de la part du destinataire.

Plan

Mise en gage (*commitment scheme*)

1. Un tiers intervient dans le protocole pour garantir l'impossibilité d'utiliser les canaux subliminaux. Il choisi une fonction de hachage cryptographique H et un nombre n.
2. On répète n fois le protocole suivant :
   1. Le signataire choisit un nombre k et communique $H(k)$ au tiers.
   2. Le tiers choisit un nombre aléatoire k'.
   3. Le signataire calcule $r = \left(g^{k \oplus k'} \mod p\right) \mod q$ et communique r au tiers.
3. Le tiers choisit $n - 1$ des k et demande à ce qu'ils lui soient communiqués, il vérifie que le protocol a bien été respecté.
4. Le signataire peut utiliser le k resté secret pour signer son message.

Mise en gage (*commitment scheme*)

1. Un tiers intervient dans le protocole pour garantir l'impossibilité d'utiliser les canaux subliminaux. Il choisi une fonction de hachage cryptographique H et un nombre n.
2. On répète n fois le protocole suivant :
   1. Le signataire choisit un nombre k et communique $H(k)$ au tiers.
   2. Le tiers choisit un nombre aléatoire k'.
   3. Le signataire calcule $r = \left(g^{k \oplus k'} \mod p\right) \mod q$ et communique r au tiers.
3. Le tiers choisit $n - 1$ des k et demande à ce qu'ils lui soient communiqués, il vérifie que le protocol a bien été respecté.
4. Le signataire peut utiliser le k resté secret pour signer son message.

Mise en gage (*commitment scheme*)

1. Un tiers intervient dans le protocole pour garantir l'impossibilité d'utiliser les canaux subliminaux. Il choisi une fonction de hachage cryptographique H et un nombre n.
2. On répète n fois le protocole suivant :
   1. Le signataire choisit un nombre k et communique $H(k)$ au tiers.
   2. Le tiers choisit un nombre aléatoire k'.
   3. Le signataire calcule $r = \left(g^{k \oplus k'} \mod p\right) \mod q$ et communique r au tiers.
3. Le tiers choisit $n - 1$ des k et demande à ce qu'ils lui soient communiqués, il vérifie que le protocol a bien été respecté.
4. Le signataire peut utiliser le k resté secret pour signer son message.

Mise en gage (*commitment scheme*)

1. Un tiers intervient dans le protocole pour garantir l'impossibilité d'utiliser les canaux subliminaux. Il choisi une fonction de hachage cryptographique H et un nombre n.
2. On répète n fois le protocole suivant :
   1. Le signataire choisit un nombre k et communique $H(k)$ au tiers.
   2. Le tiers choisit un nombre aléatoire k'.
   3. Le signataire calcule $r = \left(g^{k \oplus k'} \mod p\right) \mod q$ et communique r au tiers.
3. Le tiers choisit $n - 1$ des k et demande à ce qu'ils lui soient communiqués, il vérifie que le protocol a bien été respecté.
4. Le signataire peut utiliser le k resté secret pour signer son message.

Mise en gage (*commitment scheme*)

1. Un tiers intervient dans le protocole pour garantir l'impossibilité d'utiliser les canaux subliminaux. Il choisi une fonction de hachage cryptographique H et un nombre n.
2. On répète n fois le protocole suivant :
   1. Le signataire choisit un nombre k et communique $H(k)$ au tiers.
   2. Le tiers choisit un nombre aléatoire k'.
   3. Le signataire calcule $r = \left(g^{k \oplus k'} \mod p\right) \mod q$ et communique r au tiers.
3. Le tiers choisit $n-1$ des k et demande à ce qu'ils lui soient communiqués, il vérifie que le protocol a bien été respecté.
4. Le signataire peut utiliser le k resté secret pour signer son message.

Mise en gage (*commitment scheme*)

1. Un tiers intervient dans le protocole pour garantir l'impossibilité d'utiliser les canaux subliminaux. Il choisi une fonction de hachage cryptographique H et un nombre n.
2. On répète n fois le protocole suivant :
   1. Le signataire choisit un nombre k et communique $H(k)$ au tiers.
   2. Le tiers choisit un nombre aléatoire k'.
   3. Le signataire calcule $r = \left(g^{k \oplus k'} \mod p\right) \mod q$ et communique r au tiers.
3. Le tiers choisit $n - 1$ des k et demande à ce qu'ils lui soient communiqués, il vérifie que le protocol a bien été respecté.
4. Le signataire peut utiliser le k resté secret pour signer son message.

Mise en gage (*commitment scheme*)

1. Un tiers intervient dans le protocole pour garantir l'impossibilité d'utiliser les canaux subliminaux. Il choisi une fonction de hachage cryptographique H et un nombre n.
2. On répète n fois le protocole suivant :
   1. Le signataire choisit un nombre k et communique $H(k)$ au tiers.
   2. Le tiers choisit un nombre aléatoire k'.
   3. Le signataire calcule $r = \left(g^{k \oplus k'} \mod p\right) \mod q$ et communique r au tiers.
3. Le tiers choisit $n - 1$ des k et demande à ce qu'ils lui soient communiqués, il vérifie que le protocol a bien été respecté.
4. Le signataire peut utiliser le k resté secret pour signer son message.

Plan

Limites des contre-mesures

- Ce protocole dispose lui-même d'un canal subliminal entre le tiers de confiance et le signataire.

Canal subliminal dans le protocole de mise en gage

Les k sont choisis de manière à ce qu'on puisse appliquer la méthode des résidus quadratiques à $H(k)$.

- Tout protocole naïf visant à garantir l'absence d'un canal subliminal dispose en réalité lui-même d'un canal subliminal de 1 bit.

Canal subliminal dans tout protocole de contre-mesure

La réussite du protocole vaut 1, l'échec du protocole vaut 0.

Limites des contre-mesures

- Ce protocole dispose lui-même d'un canal subliminal entre le tiers de confiance et le signataire.

Canal subliminal dans le protocole de mise en gage

Les k sont choisis de manière à ce qu'on puisse appliquer la méthode des résidus quadratiques à $H(k)$.

- Tout protocole naïf visant à garantir l'absence d'un canal subliminal dispose en réalité lui-même d'un canal subliminal de 1 bit.

Canal subliminal dans tout protocole de contre-mesure

La réussite du protocole vaut 1, l'échec du protocole vaut 0.

Limites des contre-mesures

- Ce protocole dispose lui-même d'un canal subliminal entre le tiers de confiance et le signataire.

Canal subliminal dans le protocole de mise en gage

Les k sont choisis de manière à ce qu'on puisse appliquer la méthode des résidus quadratiques à $H(k)$.

- Tout protocole naïf visant à garantir l'absence d'un canal subliminal dispose en réalité lui-même d'un canal subliminal de 1 bit.

Canal subliminal dans tout protocole de contre-mesure

La réussite du protocole vaut 1, l'échec du protocole vaut 0.

Limites des contre-mesures

- Ce protocole dispose lui-même d'un canal subliminal entre le tiers de confiance et le signataire.

Canal subliminal dans le protocole de mise en gage

Les k sont choisis de manière à ce qu'on puisse appliquer la méthode des résidus quadratiques à $H(k)$.

- Tout protocole naïf visant à garantir l'absence d'un canal subliminal dispose en réalité lui-même d'un canal subliminal de 1 bit.

Canal subliminal dans tout protocole de contre-mesure

La réussite du protocole vaut 1, l'échec du protocole vaut 0.

Conclusion

Il ne faut *jamais* faire confiance à une implémentation de DSA que l'on ne peut pas auditer.

Questions