Le Chiffrement Asymétrique et la Sécurité Prouvée

David Pointcheval

Habilitation à Diriger des Recherches
Université Paris VII - Denis Diderot

École normale supérieure

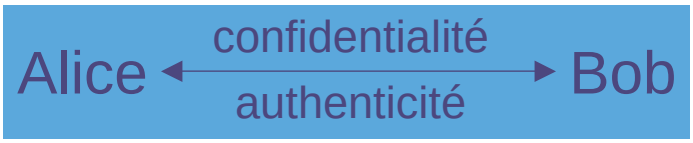
Sommaire

Sommaire

1. Le chiffrement asymétrique
2. Les hypothèses algorithmiques
3. Les preuves de sécurité
4. Un exemple : OAEP
5. La sécurité pratique
6. Conclusion

Deux clés…

Cryptographie asymétrique



Diffie-Hellman 1976

Chiffrement asymétrique :

Bob possède un couple de « clés »

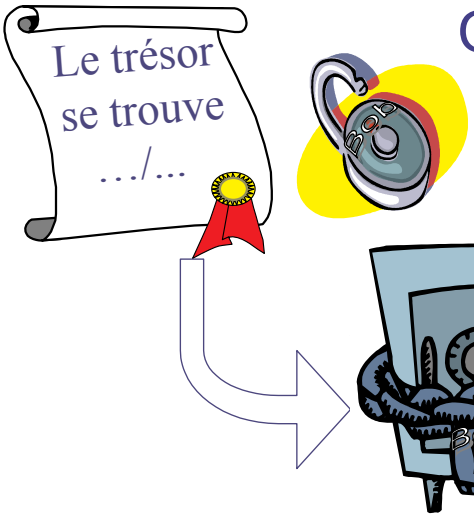
– une clé publique (de chiffrement k_e)

qui permet à qui le souhaite de lui chiffrer un message ⇒ connue de tous (dont Alice)

– une clé privée (de déchiffrement k_d)

qui lui permet de déchiffrer ⇒ connue de Bob uniquement

Chiffrement / déchiffrement décryptement



Grâce à la clé publique de Bob,
Alice peut fermer un coffre,
avec le message à l'intérieur
(*chiffrer le message*)

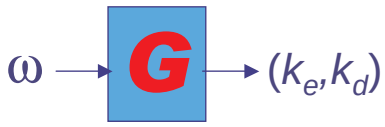
sauf Bob, avec
sa clé privée
(*il peut déchiffrer*)

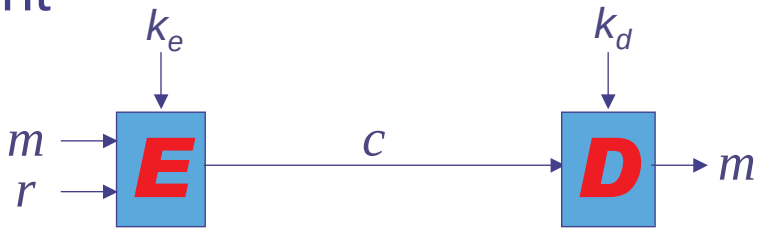
Alice envoie à Bob ce coffre
que nul ne peut ouvrir
(*impossible de décrypter*)

Un schéma de chiffrement

3 algorithmes :

- ***G*** - génération des clés
- ***E*** - chiffrement
- ***D*** - déchiffrement





Confidentialité = impossibilité de retrouver m
à partir de c sans la clé privée k_d

Confidentialité calculatoire

Le chiffré est calculé par $c = \boldsymbol{E}_{k_e}(m;r)$

- la clé k_e est publique
- un unique m satisfait cette relation
  (avec éventuellement plusieurs r)

Au moins la recherche exhaustive sur m et r permet de retrouver m, peut-être mieux !

$\Rightarrow$ confidentialité inconditionnelle impossible

hypothèses algorithmiques

Sommaire

Factorisation entière et RSA

- Multiplication/Factorisation :
  - $p, q \mapsto n = p.q$ facile (quadratique)
  - $n = p.q \mapsto p, q$ difficile (super-polynomial)

Fonction à sens-unique

- Fonction RSA, de $\mathbf{Z}_n$ dans $\mathbf{Z}_n$ (avec $n=pq$) pour un exposant e fixé — Rivest-Shamir-Adleman 1978
  - $x \mapsto x^e \bmod n$ facile (cubique)
  - $y=x^e \bmod n \mapsto x$ difficile (sans p ni q)

Problème RSA

$x = y^d \bmod n$ où $d = e^{-1} \bmod \varphi(n)$

trappe

Factorisation entière et RSA

- Multiplication/Factorisation :
  - $p, q \mapsto n = p.q$ facile (quadratique)
  - $n = p.q \mapsto p, q$ difficile (super-polynomial)

Fonction à sens-unique

- Fonction RSA, de $\mathbf{Z}_n$ dans $\mathbf{Z}_n$ (avec $n=pq$) pour un exposant e fixé — Rivest-Shamir-Adleman 1978
  - $x \mapsto x^e \bmod n$ facile (cubique)
  - $y=x^e \bmod n \mapsto x$ difficile (sans p ni q)

Problème RSA

$x = y^d \bmod n$ où $d = e^{-1} \bmod \varphi(n)$

trappe

chiffrement

Factorisation entière et RSA

- Multiplication/Factorisation :
  - $p, q \mapsto n = p.q$ facile (quadratique)
  - $n = p.q \mapsto p, q$ difficile (super-polynomial)

Fonction à sens-unique

- Fonction RSA, de $\mathbf{Z}_n$ dans $\mathbf{Z}_n$ (avec $n=pq$) pour un exposant e fixé — Rivest-Shamir-Adleman 1978
  - $x \mapsto x^e \bmod n$ facile (cubique)
  - $y=x^e \bmod n \mapsto x$ difficile (sans p ni q)

Problème RSA

$x = y^d \bmod n$ où $d = e^{-1} \bmod \varphi(n)$

trappe

décryptement difficile

Factorisation entière et RSA

- Multiplication/Factorisation :
  - $p, q \mapsto n = p.q$ facile (quadratique)
  - $n = p.q \mapsto p, q$ difficile (super-polynomial)

Fonction à sens-unique

- Fonction RSA, de $\mathbf{Z}_n$ dans $\mathbf{Z}_n$ (avec $n=pq$) pour un exposant e fixé — Rivest-Shamir-Adleman 1978
  - $x \mapsto x^e \bmod n$ facile (cubique)
  - $y=x^e \bmod n \mapsto x$ difficile (sans p ni q)

Problème RSA

$x = y^d \bmod n$ où $d = e^{-1} \bmod \varphi(n)$

trappe

clé déchiffrement

Variantes et autres problèmes

- RSA
  - Flexible : $(y, n) \mapsto (x, e), y = x^e \bmod n$
  - Relié : (*Dependent-RSA*) (P EC-1999)
    pour n et e fixés, $y=x^e \bmod n \mapsto (x+1)^e \bmod n$
- Logarithme discret : $g, y=g^x \mapsto \log_g(y) = x$
- Diffie-Hellman
  - Calcul : $(A=g^a, B=g^b) \mapsto \mathrm{DH}(A,B)=g^{ab}$
  - Décision : $(A=g^a, B=g^b, C=g^c) \mapsto C \stackrel{?}{=} \mathrm{DH}(A,B)$
  - Gap : Gap-Problems (OP PKC-2001)

    Résoudre C-DH à l'aide d'un oracle D-DH

Estimations de complexité

Estimations pour la factorisation Lenstra-Verheul 2000

Module (en bits)	Mips-Year (en $\log_2$)	Opérations (en $\log_2$)	
512	13	58	Record Août 1999
1024	35	80	Repère
2048	66	111	
4096	104	149	
8192	156	201	

Convenables pour RSA

Bornes inférieures pour LD dans $\mathbf{Z}_p^*$

Sommaire

1. Le chiffrement asymétrique
2. Les hypothèses algorithmiques
3. Les preuves de sécurité
4. Un exemple : OAEP
5. La sécurité pratique
6. Conclusion

Hypothèse algorithmique *nécessaire*

- $n=pq$: module **public**
  e : exposant **public**
- $d=e^{-1} \bmod \varphi(n)$: **privé**

Chiffrement RSA

$\mathbf{E}(m) = m^e \bmod n$

$\mathbf{D}(c) = c^d \bmod n$

Si le problème RSA est facile,
clairement, la confidentialité n'est pas garantie :
n'importe qui peut retrouver m à partir de c

Hypothèse algorithmique *suffisante ?*

Les preuves de sécurité garantissent que l'hypothèse est ***suffisante*** pour la confidentialité :

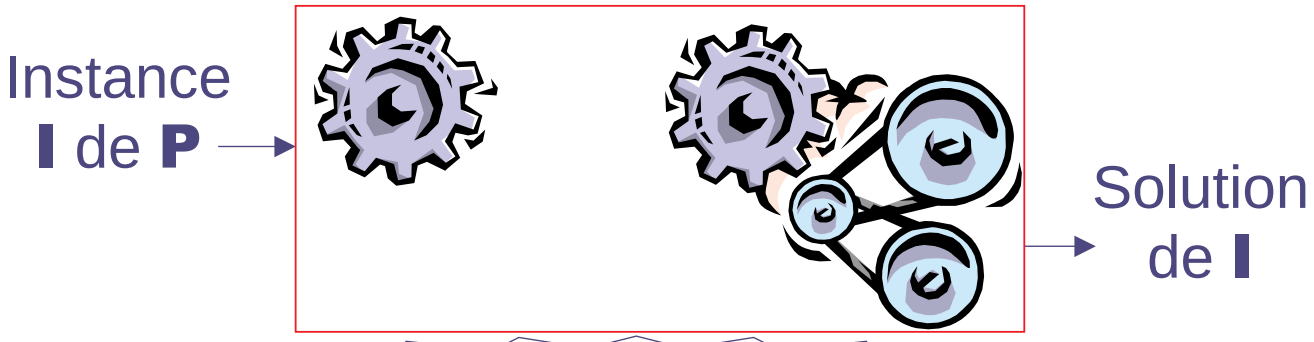
- si un adversaire parvient à violer la confidentialité
- on peut mettre en défaut l'hypothèse

⇒ preuve par réduction

Preuve par réduction

Réduction d'un problème **P** à une attaque *Atk* :

- Soit un attaquant *A* qui parvient à son but alors *A* peut être utilisé pour résoudre **P**



P insoluble ⇒ schéma incassable

Protocole prouvé sûr

Pour prouver la sécurité d'un protocole cryptographique, on doit

- préciser les hypothèses algorithmiques
- préciser les notions de sécurité à garantir
- présenter une réduction :
  un attaquant permet de contredire les hypothèses

Notions de sécurité

En fonction des besoins, on définit

- les objectifs de l'adversaire
- les moyens,
  soit les informations mises à sa disposition.

Confidentialité élémentaire

- Non-inversibilité (OW - One-Wayness) :

  sans la clé privée, il est calculatoirement impossible de retrouver le message clair

$$\mathrm{Succ}^{ow}(\mathcal{A}) = \Pr_{m,r}\left[\mathcal{A}(c) = m \middle| c = \mathbf{E}(m;r)\right]$$

Insuffisant si on a déjà de l'information sur m :

- « Message au sujet de XXXXX »
- « Ma réponse est XXX »

Confidentialité forte

- Sécurité sémantique (IND - Indistinguishability) :

  GM 1984

  le chiffré ne révèle aucune *autre* information sur le message clair à un **adversaire polynomial**

$$\mathrm{Adv}^{ind}(\mathcal{A}) = 2\Pr_{r,b}\left[\mathcal{A}_2(m_0,m_1,c,s) = b \middle| \begin{array}{c}(m_0,m_1,s) \leftarrow \mathcal{A}_1(k_e)\\ c \leftarrow \mathbf{E}(m_b,r)\end{array}\right] - 1$$

Non-malléabilité

- Non-malléabilité (NM - Non-Malleability) :

DDN 1991

Aucun adversaire polynomial ne peut dériver de $c=\mathbf{E}(m;r)$ un deuxième chiffré $c'=\mathbf{E}(m';r')$, de façon à ce que les clairs m et m' soient reliés

non-malléabilité

$\Downarrow$

sécurité sémantique

$\Downarrow$

non-inversibilité

Attaques de base

- Attaques à clairs choisis (CPA - Chosen-Plaintext Attacks)

Dans l'environnement à clé publique, l'adversaire peut chiffrer tout message de son choix, grâce à la clé publique

$\Rightarrow$ attaque de base

- Autres informations : accès à des oracles
  - attaque par réaction : c valide ?
  - attaque par vérification : $(m,c) \mapsto m \stackrel{?}{=} \mathbf{D}(c)$

Attaques à chiffrés choisis

- Attaques à chiffrés choisis
  (CCA - Chosen-Ciphertext Attacks)

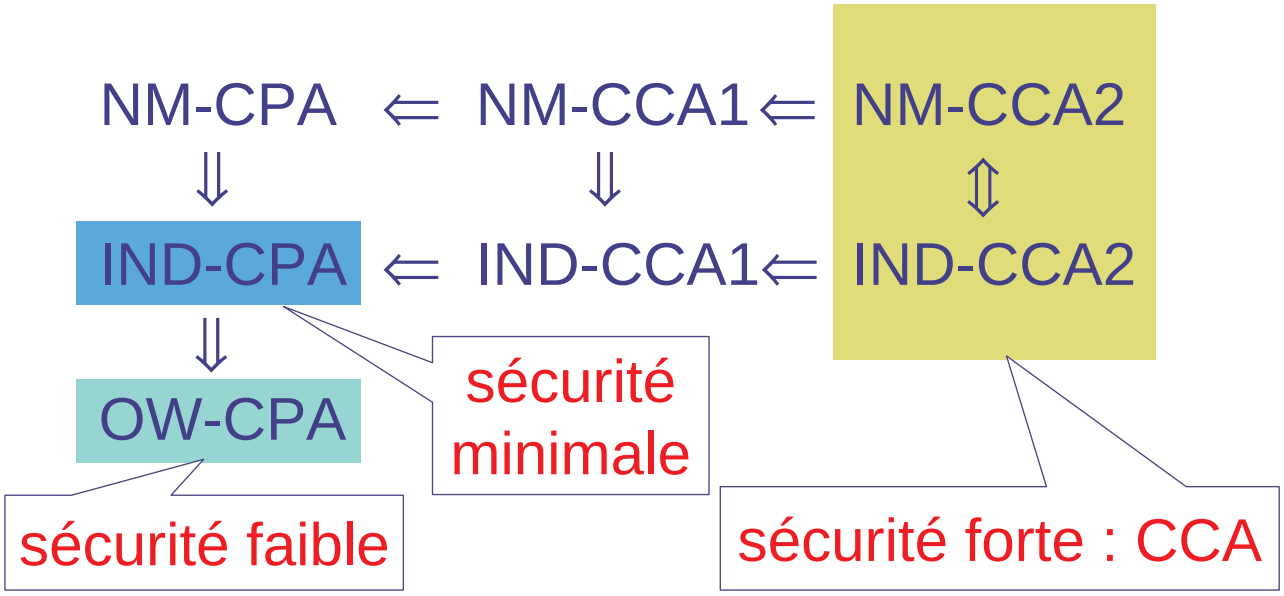
  L'adversaire a accès à l'oracle de déchiffrement
  soit le clair de tout chiffré de son choix
  (sauf le challenge)

  – non-adaptatives (CCA1) NY 1990
  accès avant de recevoir le challenge

  – adaptatives (CCA2) RS 1991
  accès illimité

Relations BDPR C-1998

Implications et séparations



Sommaire

- Le chiffrement asymétrique
- Les hypothèses algorithmiques
- Les preuves de sécurité
- Un exemple : OAEP
- La sécurité pratique
- Conclusion

Une permutation à trappe

Une permutation à sens-unique à trappe conduit à un schéma OW-CPA

Ex : RSA

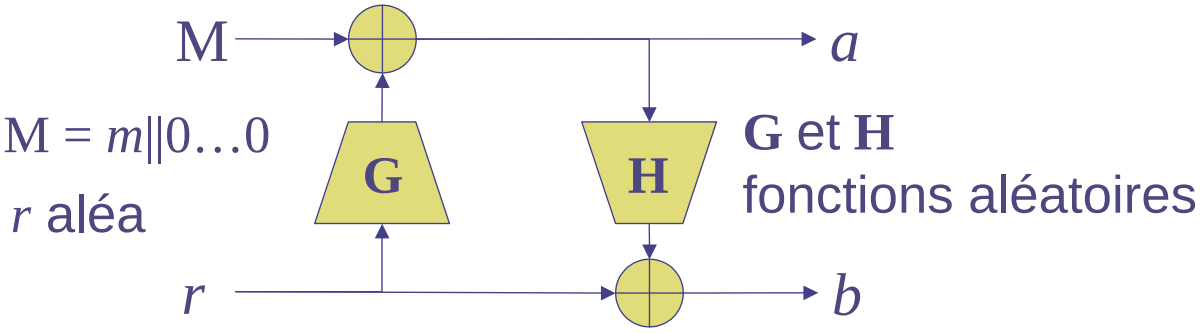
$$f(m) = m^e \bmod n$$

$$g(c) = c^d \bmod n$$

Mais niveau de sécurité insuffisant !

On veut la sécurité forte : IND-CCA2

OAEP Bellare-Rogaway 1994



E(m) : Calculer a,b puis retourner $c=f(a||b)$
D(c) : Calculer $a||b = g(c)$
inverser OAEP, et retourner m
(*si la redondance est satisfaite*)

Oracle aléatoire

- Aucun schéma vraiment **efficace** n'admet de preuve de **sécurité forte** par réduction
- Aucun schéma inefficace n'a d'intérêt pratique (*sécurité transparente*)

⇒ hypothèse supplémentaire

Ex : modèle de l'oracle aléatoire (ROM)
Bellare-Rogaway 1993

certaines fonctions (**G** et **H**)
sont considérées parfaitement aléatoires

OAEP (suite)

Dans le modèle de l'oracle aléatoire, OAEP

- conduit à un schéma IND-CPA à partir de toute *permutation à sens-unique à trappe*
- et CCA ?
  - admis jusqu'à très récemment
    ⇒ RSA-OAEP retenu par
    RSA PKCS, SET, IETF, IEEE, ISO, …
  - finalement faux Shoup 2000

RSA-OAEP FOPS C-2001

- OAEP conduit au niveau CCA à partir d'une *permutation à sens-unique* ***sur un domaine partiel****, à trappe* :
  - $(a,b) \mapsto f(a \,\|\, b)$ à sens-unique, à trappe
  - $f(a \,\|\, b) \mapsto a$ également difficile
- RSA-Partiel ⇔ RSA

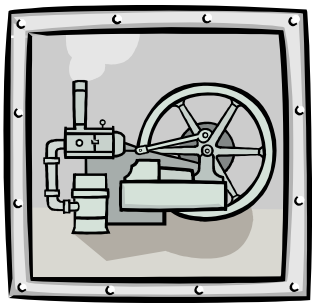
  avec un (t, ε)-oracle qui extrait a de $(a \,\|\, b)^e \bmod n$,
  RSA(n,e) résolu avec probabilité ε^2 en temps $2t$
- IND-CCA2 de RSA-OAEP ⇔ RSA

  Heureusement pour les applications industrielles !

Intérêt pratique

- RSA-OAEP : construction efficace, prouvée IND-CCA2 sous RSA (ROM)
- Mais la réduction est quadratique

Attaquant contre IND-CCA2 en t



Algorithme contre RSA en $t' \approx t^2$

RSA 1024 bits impose $t' > 2^{80}$ donc $t > 2^{40}$

$\Rightarrow$ sécurité prouvée en 2^{40} !

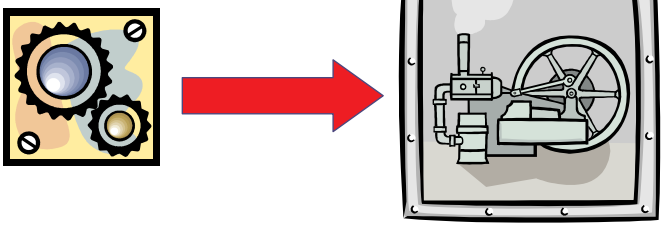
(ou 2^{74}, … mais avec 4096 bits : pas pratique)

Sommaire

- Le chiffrement asymétrique
- Les hypothèses algorithmiques
- Les preuves de sécurité
- Un exemple : OAEP
- La sécurité pratique
- Conclusion

Sécurité pratique

Attaquant en t



Algorithme contre **P** en $t' = T(t)$

- Théorie de la complexité : T polynomiale
- Sécurité exacte : T explicite
- Sécurité pratique : T petite (linéaire)

Ex : $t' = 4t$

P insoluble en moins de 2^{80} opérations

$\Rightarrow$ schéma incassable en moins de 2^{78} op.

REACT OP RSA-2001

E$(m ; r)$= $a = f(r)$ avec $r \in \mathcal{M}$

$b = k \oplus m$ où $k = \mathrm{G}(r)$

$c = \mathrm{H}(m,r,a,b)$

D(a,b,c): Calculer $r = g(a)$ et $k = \mathrm{G}(r)$

extraire $m = k \oplus b$

si $c = \mathrm{H}(m,r,a,b)$ et $r \in \mathcal{M}$

alors retourner m sinon « refus »

Conversion ***efficace*** de toute fonction injective f, de type ***Gap-Problem***, en chiffrement IND-CCA2

Sécurité pratique

$G : \mathcal{M} \rightarrow \{0,1\}^{\ell_G}$ $H : \{0,1\}^* \rightarrow \{0,1\}^{\ell_H}$ (ROM)

- Si un adversaire $\mathcal{A}$ contre IND-CCA2 obtient un avantage ε en temps t après q_G, q_H et $q_{\mathbf{D}}$ questions à G, H et **D** resp.

- alors on peut casser le Gap-Problem f avec probabilité $\frac{\varepsilon}{2} - \frac{q_{\mathbf{D}}}{2^{\ell_H}}$

  en temps $t' \leq t + (q_G + q_H)\, T_{\text{test}} \leq 2t$ (si T_{test} petit)

REACT : exemples

- Réduction linéaire : sécurité pratique
- Efficacité : version hybride
- RSA-REACT

  $\Rightarrow$ sécurité avec 1024 bits prouvée en 2^{78} !

  (à comparer à 2^{40} pour RSA-OAEP 1024 bits)
- El Gamal-REACT

  (PSEC-3 sur courbes elliptiques)

  $\Rightarrow$ sécurité basée sur le Gap Diffie-Hellman

Sommaire

- Le chiffrement asymétrique
- Les hypothèses algorithmiques
- Les preuves de sécurité
- Un exemple : OAEP
- La sécurité pratique
- Conclusion

La sécurité prouvée

Trois étapes primordiales :

- notions formelles de sécurité
  - formalisation BDPR C-1998
  - nouvelles notions BBDP AC-2001
- hypothèses algorithmiques précises
  - Dependent RSA P EC-1999
  - Gap Problems OP PKC-2001
- preuves par réduction
  - REACT OP RSA-2001
  - OAEP FOPS C-2001

Nouveautés

- Étude du chiffrement
  après l'authentification — Thèse de doctorat
- Réduction efficace
  ⇒ sécurité pratique
- Nouveau formalisme — Shoup
  preuves plus claires et plus modulaires
  ⇒ affiner ce formalisme
  ⇒ prouver des protocoles complets
  (protocoles de groupes)