Cryptographie Asymétrique : Description d'un nouveau cryptogramme

Rémy Aumeunier
remy.aumeunier@libertysurf.fr

Amateur

Résumé En cryptographie asymétrique, l'un des principaux défis consiste à définir un point d'arrêt. Le schéma qui suit est basé sur une construction mathématique qui n'est pas utilisée, à ma connaissance, dans ce contexte.Cette approche a été publié sur un forum spécialisé et étudié par certain contributeur habituel voir newsgroup Usenet : sci.crypt

1 Introduction

Le chiffrement asymétrique est apparu en 1976[1], avec la publication d'un ouvrage sur la cryptographie publié par Whitfield Diffie et Martin Hellman ; mais aussi par $RalphMerkle$[2] à la même époque. Le cryptosystème asymétrique utilise 2 clefs : une clef publique et une clef privée, ou secrète. Lorsque 2 personnes (nommées par convention Alice et Bob) veulent échanger des informations via un canal ouvert ou publique, Alice publie une clef publique, Bob code son message avec la clef public d'Alice et met à disposition d'Alice le résultat du chiffrement. Puis Alice avec sa clef privée récupère les informations codées par Bob.

1.1 État de l'art

Les algorithmes de cryptographie asymétrique peuvent être regroupés en 4 grandes familles. Les plus connus sont les cryptogrammes de type RSA. Cet algorithme a été décrit en 1977 par Ronald Rivest, Adi Shamir et Leonard Adleman. Il y a aussi les cryptogrammes à courbe elliptique proposés de manière indépendante, par $NealKoblitz$[4] et $VictorMiller$[3] en 1985. Une courbe elliptique est un cas particulier de courbe algébrique avec laquelle on peut faire une addition, ce qui permet de définir un échange de clefs de type Diffie-Hellman. Il y a aussi le chiffrement El Gamal qui est un algorithme de cryptographie asymétrique fondé sur le problème du logarithme discret créé par l'Égyptien Taher Elgamal, doctorant de l'université de Stanford. puis pour finir il y a aussi plusieurs cryptosystème basé sur le célèbre problème du sac à dos.

1.2 Préambule

Quelques relations mathematiques

$$a^6 - b^6 = (a-b).(a^5 + a^4.b + a^3.b^2 + a^2.b^3 + a.b^4 + b^5)$$

$$b^n < (a-b) \qquad a^n \equiv b^n mod(a-b)$$

$$x = b + c \qquad y = b \qquad n = 6$$

$$x^n - y^n = (x-y).(x^5 + x^4y + x^3.y^2 + x^2.y^3 + x.y^4 + y^5)$$

$$a^n/(a-y) \equiv (x^5 + x^4.y + x^3.y^2 + x^2.y^3 + x.y^4 + y^5) mod(a-x)$$

$$x^n - y^n = (x-y).(a^n/(a-y) mod(a-x))$$

$$(b+c)^6 - b^6 = ((b+c)-b).(b^5 + b^4.(b+c) + b^3.(b+c)^2 + b^2.(b+c)^3 + b.(b+c)^4 + (b+c)^5)$$

$$(b+c)^6 - b^6 = (c.(b^5 + b^4.(b+c) + b^3.(b+c)^2 + b^2.(b+c)^3 + b.(b+c)^4 + (b+c)^5)$$

$$(b+c)^6 = b^6 + (c.(b^5 + b^4.(b+c) + b^3.(b+c)^2 + b^2.(b+c)^3 + b.(b+c)^4 + (b+c)^5)$$

$$(b+c)^6 = b^6 + c.(a^n/(a-b) mod(a-(b+c)))$$

2 Présentation du schéma cryptographique

Tout le cryptosystème peut se résumer par

$$(b+c)^n < (a-b) \qquad (b+c)^n = b^n + c.(\frac{a^n}{a-b}) mod(a-(b+c))$$

Alice

$$(\frac{a^n}{a-b}) \quad , \quad (a-b)$$

Bob

$$c.(\frac{a^n}{a-b}) mod(a-b-c)$$

Alice recherche par dichotomie $(b+c)^n = b^n + c.bob$

2.1 Renforcement de la sécurite

Pour éviter de faire une hypothèse sur a^n

$$ClefPublicAlice1.ClefPublicAlice2 = \frac{a^n}{(a-b)}.(a-b)$$

Alice modifie sa ClefPublicAlice1 puisque

$$((\frac{a^n}{a-b})mod(a-b-c) + (\frac{a^m}{a-b})mod(a-b-c))mod(a-b-c)$$
$$= (\frac{a^n+a^m}{a-b})mod(a-b-c))$$

et pour éviter

$$(PublicAlice1.PublicAlice2)mod(unFacteurde \quad a^n) = 0$$
$$(\frac{a^n+a^m}{a-b}).(a-b)mod(a) = 0)$$

Alice introduit du bruit

$$(\frac{a^n+a^m}{a-b} + bruitAlice) \quad , \quad (a-b)$$

Bob aussi introduit du bruit pour renforcer la securité ,après son calcul

$$ClefPubliqueBob = (c.(\frac{a^n+a^m}{a-b} + bruitAlice).mod((a-b)-c) + bruitBob$$

puis elle recherche par dichotomie $(b-c)^n + (b-c)^m$

$$(b+c)^n + (b+c)^m = \frac{b^n + b^m + c.(\frac{a^n+a^m}{a-b})mod(a-b-c)}{bruitAlice} + c$$

cf. 5 Application numérique

3 Unicité de la solution trouvée par Alice

Pour un a, b et n donné il ne peut y avoir qu'une seule valeur pour c , à cause de $(b+c)^n$ ou par ce que b et n ne sont pas des inconnues

$$(b+c)^n = b^n + c.(\frac{a^n}{a-b})mod(a-b-c)$$

ou

$$(b+c)^n + (b+c)^m = b^n + b^m + c.(\frac{a^n+a^m}{a-b})mod(a-b-c)$$

de manière plus mathématique la simple remarque

$$si \quad b^n < (a-b) \quad alors \quad (a^n)mod(a-b) = b^n$$

cf. arithmétique modulaire suffit à garantir la bijection

4 Domaine d'application

Pour que le cryptograme fonctionne ,il faut que

$$(b+c)^n < (a-b)$$

donc a doit être grand devant $(b - Privatebob)^n$ comme par définition je ne peux pas connaître PrivateBob je peux imaginer une règle qui limite le nombre de chiffres de PrivateBob par rapport aux clefs publiques par exemple a ,posséde 10,20,30,...,100 fois plus de chiffres que c et pour la taille du bruit une première approximation peut être

$$bruitBob.privatebob < publicbob$$

cf. Application numérique

5 Application numérique

Sous linux Ctrl+Alt+T exécuter bc puis un copier /coller

```
#########Alice#########
n=5
m=7
bruitprivatealice=2^280-123456789
a=1321351561652113 21^13+3213213215615146161
b=121454784

publicalice1=((a^n+a^m)/(a-b)-bruitprivatealice)
publicalice2=a-b
######################
#########Bob#########
privatebob=157412678544574
bruitprivatebob=2^284+1234781
publicbob= privatebob*(publicalice1%(publicalice2-privatebob))+
    bruitprivatebob

######################
#########Alice#######
r=1
bornsup=publicbob
borninf=0
while(r!=0)
{
c=(borninf+bornsup)/2
r=((publicbob+b^n+b^m-(b+c)^n-(b+c)^m))/bruitprivatealice+c
if(rtmp==r){r=0}
rtmp=r
if(r<0){bornsup=c}
if(r>0){borninf=c}

}
print "privatebob=";c
######################
```

6 Etude de la sécurité Attaques possibles

6.1 L'attaque des clefs *ClefPubliqueAlice*

Alice rend publique deux valeurs

$$\frac{a^n + a^m}{(a-b)} + bruitAlice \quad , \quad (a-b)$$

pour l'instant j'ai trouvé aucune attaque

6.2 L'attaque de la *ClefPubliqueBob*

Bob rend publique une valeur $ClefPubliqueBob$

$$PrivateBob.(ClefPublicAlice1.mod(ClefPublicAlice2 - PrivateBob)) + bruitBob$$

$$ClefPubliqueBob = (c.(\frac{a^n + a^m}{a-b} + bruitAlice).mod((a-b)-c) + bruitBob$$

pour l'instant j'ai trouvé aucune attaque

6.3 L'attaque par force brute

Comme le décrit l'attaque cela consiste à essayer toutes les combinaisons possibles n'ayant pas trouvé de point d'arrêt sur les clefs ClefPublicAlice1 et ClefPublicAlice2 il ne reste que la ClefPubliqueBob donc il faut annuler le bruit et faire une factorisation.J'ai pas trouvé d'algo simple pour l'instant

7 Cryptanalyse

7.1 Attaque sur texte chiffré

Une attaque sur texte chiffré seulement consiste à retrouver la clef de déchiffrement à partir d'un ou plusieurs textes chiffrés donc

$$ClefpublicBob_1 = c_1.((\frac{a^n + a^m}{a-b} + bruitAlice).mod(a-b-c_1)) + bruitBob_1$$

$$ClefpublicBob_2 = c_2.((\frac{a^n + a^m}{a-b} + bruitAlice)mod(a-b-c_2)) + bruitBob_2$$

$$\dots$$

$$ClefpublicBob_n = c_n.((\frac{a^n + a^m}{a-b} + bruitAlice)mod(a-b-c_n)) + bruitBob_n$$

pour l'instant je n'ai trouvé aucune attaque.

7.2 Attaque sur texte clair connu

Une attaque sur texte clair connu consiste à retrouver la clef de déchiffrement à partir d'un ou plusieurs textes chiffrés, connaissant le texte en clair correspondant pour l'instant je n'ai trouvé aucune attaque.

7.3 Attaque sur texte clair choisi

Une attaque sur texte clair choisi consiste à retrouver la clef de déchiffrement à partir d'un ou plusieurs textes chiffrés, l'attaquant ayant la possibilité de générer du chiffre à partir de textes en clair pour l'instant je n'ai trouvé aucune attaque.

7.4 Attaque sur texte chiffré choisi

Une attaque sur texte chiffré choisi consiste à retrouver la clef de déchiffrement à partir d'un ou plusieurs textes chiffrés, l'attaquant ayant la possibilité de générer du chiffre à partir de textes en clair pour l'instant je n'ai trouvé aucune attaque.

8 Variation possible

Alice peut effectuer quelques variations sur

$$(\frac{a^n}{a-b})mod(a-b-c))$$

$$(\frac{a^n+a^m}{a-b})mod(a-b-c))$$

....

9 Recommandation pour une mise en oeuvre

Le system est relativement bien protégé parceque la securite repause sur un savoir et non pas sur une propriété mathematique ,ici b,m ,n ainsi que par les différents bruits introduit *bruitBob* et *bruitAlice* malgré tout il faut mieux évaluer l'impacte de la taille du bruit sur la recherche du msg de Bob
cf. Domaine d'application

10 Licence et droit de propriété intellectuelle

Domaine public ou libre de toute contrainte ou notion de propriété

11 Mise en garde

Compte tenu du fait que la création d'un schéma de cryptographie asymétrique est l'une des choses les plus difficiles auxquelles je me suis retrouvé confronté, je vous conseille vivement la plus grande prudence. Autrement dit, je ne suis pas censé réussir à ce jeu, donc ...

Références

1. ↑ W. Diffie and M.E. Hellman, Multiuser cryptographic technics, Proceedings of AFIPS National Computer Conference, 109-112, 1976
2. ↑ A.J. Menezes, P.C Van Oorschot, S.A. Vanstone, Handbook of applied cryptography, CRC Press, 1997, p. 47
3. ↑ V. Miller, « Use of elliptic curves in cryptography », dans CRYPTO, n°85, 1985.
4. ↑ Neal Koblitz, « Elliptic curve cryptosystems », dans Mathematics of Computation, n°48, 1987, p.203–209