Suites de Lucas et cryptosystéme Cramer-Shoup

El Hadji Mamadou Dia et Mouhamadou Wagne

UFR Sciences Appliquées et Technologies
Universite Gaston Berger de Saint Loius
Master I Cryptologie,Codage et Application (CCA)

Decembre 2021

Étude détaillée

Définition

Soit A un anneau (unitaire) commutatif et $p \in A$. On définit la suite de Lucas $v : N \to A$ par les égalités:

$$v_0 = 2, v_1 = p, v_{n+2} = p \times v_{n+1} - v_n$$

On pose $D = p^2 - 4 \in A$ (discriminant de la suite v).

Propriétés

- Soit $0 \leqslant m \leqslant n$; il vient:

$$v_{n+m} = v_n \times v_m - v_{n-m}$$

ce qui fournit les égalités:

$$v_{2n-1} = v_n \times v_{n-1} - p,\ v_{2n} = v_n^2 - 2,\ v_{2n+1} = v_{n+1} \times v_n - p$$

- Signalons aussi l'égalité:

$$(v_n - p)^2 = (v_{n-1} - 2)(v_{n+1} - 2)$$

ainsi surtout que la relation fondamentale:

$$v_{nm} = v_n \circ v_m$$

soit encore, pour tout triplet $(n, m, p), v_{nm}(p) = v_n(v_m(p))$, qui traduit l'identité immédiate $A^{n \times m} + B^{n \times m} = (A^m)^n + (B^m)^n$.

Calcul de la suite de Lucas

Les relations ci-dessus conduisent évidemment á un algorithme de calcul de $v_n(p)$, de complexité $\bigcirc(logn)$, explicité dans le cas $A = Z$ par la pseudo–procédure Sagemath ci–dessous.

Une variante de cette procédure conduit au calcul, ‘ toujours en $\bigcirc(logn)$, du reste modulo N de $v_n(p)$, donc de $v_n(p)$ dans le cas

$$A = Z_N = Z/NZ$$

Une procédure Sagemath pour $v_n(p)$

Entrée [21]:

```
#-------------------------------------------------------------#
#Membres du groupe : El Hadji Mamadou Dia & Mohamadou Wagne
#Universite : Gaton Berger de Saint Louis
#UFR : SAT
#Parcours : Cryptotologie, Codage et Application(CCA)
#Niveau : Master I
#Module : Introduction a la Cryptographie
#Annee Universitaire : 2020/2021
#-------------------------------------------------------------#

print("_______________________________________________________")
print("----------- |Suite de Lucas| ---------------")
print("_______________________________________________________")
print("----------------- |Calcul| --------------------")
print("_______________________________________________________")
def lucas(n, p):
    if p == 0: return 1
    return n * lucas(n, p-1)
    if p & 1:
        return n * lucas(n, p-1)
    else:
        return lucas(n, p//2) ** 2
lucas(15,6)
```

```
_______________________________________________________
----------- |Suite de Lucas| ---------------
_______________________________________________________
----------------- |Calcul| --------------------
_______________________________________________________
```

Out[21]: 11390625

Théoréme fondemental

- Si dans l'une des suites récurrentes (v_n) (de Lucas), le terme v_{p-1} est divisible par p, sans qu'aucun des termes de la suite dont le rang est un diviseur de $p-1$ le soit (dans la pratique, le calcul des termes de la suite se fera modulo p), le nombre p est premier.
- Si v_{p+1} est divisible par p sans qu'aucun des termes de la suite dont le rang est un diviseur de $p+1$ le soit, p est premier.

Principe

- Le principe de ce genre d'algorithme ne repose que sur une hypothèse d'intractabilité standard, à savoir, la dureté de la problème de décision Diffie-Hellman dans le groupe sous-jacent(groupes de premier ordre).
  - Cette fonction est l'emploi d'un hachage universel unidirectionnel et effectue des calculs supplémentaires, conduisant à un texte chiffré.
- Le principe de ce genre d'algorithme peut être fondé sur la décision de Paillier.
- Il peut aussi etre fondé sur l'hypothèse (tout à fait classique) de résiduosité quadratique (QR).
  - l'algorithme basée sur le résiduosité quadratique (QR) de la généralisation n'est pas trop efficace dans la pratique.

Concept

- Une clé publique **Kpub** $= (g_1, g_2, c, d, h, H)$ pour chiffrer.
- Une clé privée **Kpriv** $= (x_1, x_2, y_1, y_2, z)$ pour déchiffrer.
- Propriété : La connaissance de **Kpub** ne permet pas de déduire **Kpriv**.
- **DKpriv (EKpub (M)) = M**.
- La résistance aux attaques adaptatives à chiffrés choisis (IND-CCA2).

Avantage

- Il est résistant aux attaques adaptatives à chiffrés choisis (IND-CCA2).
- Il est prouvé sûr dans le modèle standard.
- Il est efficace.

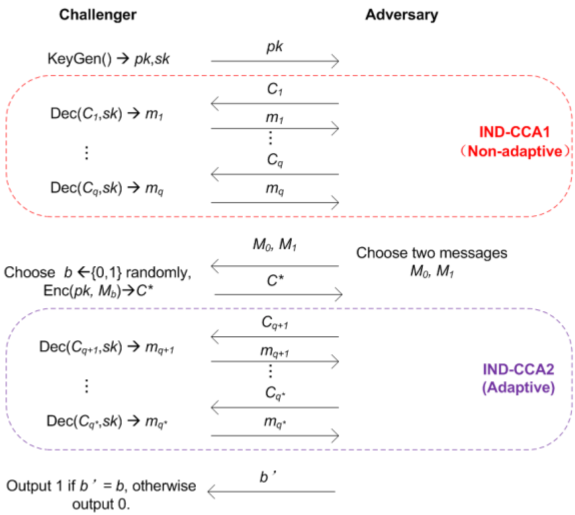
Fonctionnement

- Cependant, à l'inverse du chiffrement symétrique où le nombre de clés est le problème majeur, ici, seules n paires sont nécessaires.
- Alors, chaque utilisateur possède une paire (Kpub, Kpriv) et tous les transferts de message ont lieu avec ces clés.
- Chaque utilisateur conserve sa clé secrète (privée) sans jamais la divulguer. Seule la clé publique devra être distribuée.

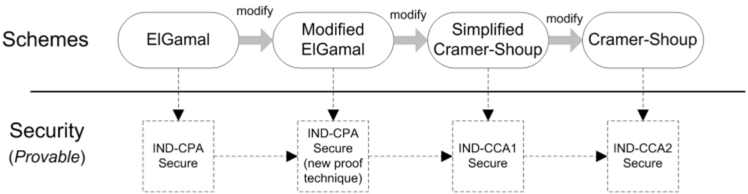
Probléme Mathématiques

- Le cryptosystéme Cramer-Shoup se base sur des généralisations naturelles de groupes cycliques.

Modéle de Sécurité



Schéma


Schemes
ElGamal
modify
Modified ElGamal
modify
Simplified Cramer-Shoup
modify
Cramer-Shoup
Security
(Provable)
IND-CPA Secure
IND-CPA Secure (new proof technique)
IND-CCA1 Secure
IND-CCA2 Secure