UCBL – MIF26 **Juin 2021**
Crypto
Nom :
Prénom :

Examen

(Durée 1h20)

Dans toute la suite, on supposera que Alice génère une paire de clés RSA $R.pk=\{n=pq,e=17\}$ et $R.sk=\{d\}$ et une paire de clés Paillier $P.pk=\{n\}$ et $P.sk=\{\rho\}$ de taille 2048 (n s'écrit avec 2048 bits). On notera $\phi(n)$ la fonction d'Euler. On considèrera, en outre, le protocole Scal défini comme suit :

Protocole Scal. Supposons que Alice et Bob disposent chacun de 30 bits secrets, respectivement $a_1,\ldots,a_{30}$ et $b_1,\ldots,b_{30}$. Alice souhaite connaitre $\sum a_i b_i$. On note E l'ensemble défini par $E=\{i \mid i\in\{1,\ldots,30\} \,;\, b_i=1\}$. Pour réaliser ceci, on propose le protocole P défini par :

1 – Alice génère 30 encryptions $A_1,\ldots,A_{30}$ de $a_1,\ldots,a_{30}$ avec la fonction Paillier.Encrypt et les envoie à Bob.

2 – Bob calcule $X=\prod_{i\in E} A_i \bmod n^2$.

3 – Bob calcule $Y=Paillier.Encrypt(P.pk,0)$, $X\leftarrow X\times Y \bmod n^2$ et envoie X à Alice

4 –Alice retourne $Paillier.Decrypt(P.sk,X)$

Parmi les affirmations suivantes, dire (en justifiant concisément votre réponse) celles qui sont vraies et celles qui sont fausses (aucun point négatif…donc répondre à toutes les questions).

1 – $\log_2 n<150$

Faux. $\log_2 n\approx 2048$

2 – $pgcd(\phi(n),d)=1$

Vrai, car d est inversible modulo $\phi(n)$. C'est e son inverse

3 – Soient x,y,r,s des éléments de $(\mathbf{Z}/n\mathbf{Z})^*$, $M\leftarrow Paillier.Encrypt(P.pk,(x+r)\times(y+s) \bmod n)$, $X\leftarrow Paillier.Encrypt(P.pk,x)$, $Y\leftarrow Paillier.Encrypt(P.pk,y)$ et $W\leftarrow Paillier.Encrypt(P.pk,r\times s \bmod n)$.

$$Paillier.Decrypt(P.sk,\ M\times X^{-s}\times Y^{-r}\times W^{-1} \bmod n^2)=x\times y \bmod n$$

Vrai. A vérifier avec propriétés homomorphes Paillier

4 – Il existe un algorithme A rapide (complexité polynomiale) tel que $A(n)$ retourne $\phi(n)$.

Faux. Sous réserve qu'il n'existe pas d'algorithme polynomial de factorisation (voir section reduction polynomiale du cours)

5 – Alice choisit aléatoirement un message $m\in\{1,\ldots,2^{100}\}$, l'encrypte avec RSA, i.e calcule $M=RSA.Encrypt(R.pk,m)$ et envoie M à Bob. Bob peut retrouver m rapidement à coup sûr (avec une probabilité de 1) en temps raisonnable.

Vrai. Car $e=17$. Donc $m=M^{1/17}$ (voir exo voyante australienne)

6 – Soit a,e,n trois entiers et $ExpMod$ l'algorithme suivant :

<u>ExpMod(a,b,c)</u>

$r=1$

pour i=1 à b

r=r×a mod c

retourner r

ExpMod est un algorithme d'exponentiation modulaire (qui calcule $a^e \bmod n$) qui peut être utilisé dans la fonction de **chiffrement** *RSA.Encrypt*.

Vrai. Complexité exponentielle (car boucle de e itérations) mais $e=17$ ici donc ok pour le chiffrement…pas ok pour le déchiffrement

7 – Soit $c=RSA.Encrypt(R.pk,m)$. On a $RSA.Decrypt(R.sk,\ c^a \bmod n)= a\times m \bmod n$

Faux. $RSA.Decrypt(R.sk,\ c^a \bmod n)= m^a \bmod n$

8 – La table de hachage $h : \mathbf{N}\rightarrow Z/nZ$ définie par $h(x)=x^e \bmod n$ est résistante aux collisions (il est difficile de trouver deux entiers m_1,m_2 (même plus grands que n) tels que $h(m_1)=h(m_2)$).

Faux. $h(m)=h(m+n)$

9 – La fonction $f : (Z/35Z)^*\rightarrow (Z/35Z)^*$ définie par $f(x)=x^{24} \bmod 35$ est une bijection.

Faux. comme $24=\phi(35)$, $f(x)=1$ pour tout x dans $(Z/35Z)^*$

10 – $Paillier.Decrypt(P.sk,1)$=1

Faux. $Paillier.Decrypt(P.sk,1)$=0

11 – La fonction $f : (Z/nZ)^* \rightarrow (Z/nZ)^*$ définie par $f(x)=x^{\phi(n)-1} \bmod n$ est une bijection.

Vrai. $f(x)=x^{-1} \bmod n$

12 – La fonction $f : (Z/nZ)^*\rightarrow (Z/nZ)^*$ définie par $f(x)=(x^d)^e \bmod n$ est une bijection.

Faux. Par construction de RSA, $f(x)=x$

13 – La fonction de chiffrement *Paillier.Encrypt* est probabiliste.

Vrai. Un nombre aléatoire r est choisi lors de chaque encryption, i.e. un grand nombre d'encryptions possibles pour un même message.

14 – Soient $c=Paillier.Encrypt(P.pk,m)$ et $c'=Paillier.Encrypt(P.pk,m')$. L'élément $c.c' \bmod n^2$ est un chiffrement de $m+m' \bmod n$

Vrai. propriétés d'homomorphie de Paillier

15 – Soient $m_0,m_1 \in Z/nZ$ deux messages arbitraires publiques. Bob choisit un bit $b\in\{0,1\}$ et envoie une encryption $M= RSA.Encrypt(R.pk,m_b)$ à Alice. Oscar, qui contrôle le réseau, peut transformer M en M' tel que $RSA.Decrypt(R.sk,M')=m_{1-b}$.

Vrai. (Voir TD)

16 – Même question que la précédente en considérant Paillier au lieu de RSA

Vrai. (Voir TD)

17 – La valeur retournée par Alice est correcte si Alice et Bob respectent le protocole Scal

Vrai. Valeur retournée est égale à $\sum_{i\in E} a_i \bmod n=\sum_{i\in E} a_i=\sum a_ib_i$

18 – Bob peut retrouver (en temps raisonnable) $a_1,\dots,a_{30}$ en déviant éventuellement du protocole (Alice est supposée le respecter)

Vrai ou Faux…dépend de la justification. Si dévier du protocole consiste à choisir des b_i non binaires alors oui, sinon non. S'il choisit $b_i=2^i$ (par exemple) alors il retrouve tout les a_i.

19 – Alice peut retrouver (en temps raisonnable) $b_1,\dots,b_{30}$ en déviant du protocole (Bob est supposé le respecter)

Comme question 18 en intervertissant Bob et Alice.

20 – Remplaçons l'étape 3 du protocole Scal par « *Bob envoie X à Alice* » (autrement dit, on ne fait pas $X \leftarrow X \times Y \bmod n^2$). Supposons qu'Alice respecte ce (nouveau) protocole (e.g. $A_1, \ldots, A_{30}$ sont des encryptions de bits). Elle peut retrouver (en temps raisonnable) $b_1, \ldots, b_{30}$.

Vrai. Par force brute. Elle essaie tous les produits $X = \prod_{i \in E} A_i \bmod n^2$ pour tous les ensembles E possibles et compare avec Y, i.e. si $X = Y$ alors elle a retrouvé E