

# Théorème des Restes Chinois et Paramètres Cryptographiques

Alexandre Duc

1. Choix de la Taille des Clefs
2. Théorème des Restes Chinois
3. Racines Carrées
4. Génération de Paramètres Cryptographiques

### Question

Comment choisir la taille d'une clef symétrique ? D'une clef asymétrique ?

# Choix de la Taille des Clefs

- **Clef symétrique** : complexité du bruteforce. Loi de Moore.
- **Clef asymétrique** : en plus de la loi de Moore, considérer la **difficulté supposée** de casser un problème difficile à l'aide des meilleurs algorithmes connus.
- Problèmes difficiles sous-jacents à RSA et El Gamal :
  1. **Factorisation** de  $n = pq$
  2. **Logarithme discret** dans  $\mathbb{Z}_p^*$  ou sur une courbe elliptique.

# Factorisation

## Quelques Records

Problème	Taille (bits)	Année	Auteur(s)
RSA-100	330	1991	A. K. Lenstra
RSA-110	364	1992	A. K. Lenstra et M. S. Manasse
RSA-120	397	1993	T. Denny
<b>RSA-129</b>	<b>426</b>	<b>1994</b>	<b>A. K. Lenstra et al.</b>
RSA-130	430	1996	A. K. Lenstra et al.
RSA-140	463	1999	H. te Riele et al.
RSA-150	496	2004	K. Aoki et al.
<b>RSA-155</b>	<b>512</b>	<b>1999</b>	<b>H. te Riele et al.</b>
RSA-160	530	2003	J. Franke et al.
RSA-170	563	2009	D. Bonenberger and M. Krone
RSA-180	596	2010	S. A. Danilov et I. A. Popovyan
RSA-190	629	2010	A. Timofeev et I. A. Popovyan
<b>RSA-193</b>	<b>640</b>	<b>2005</b>	<b>J. Franke et al.</b>
RSA-210	696	2013	R. Propper
RSA-232	768	2009	T. Kleinjung et al.
<b>RSA-240</b>	<b>795</b>	<b>2019</b>	<b>F. Bourdot et al.</b>
<b>RSA-250</b>	<b>829</b>	<b>2020</b>	<b>F. Bourdot et al.</b>

# Logarithme Discret

## Records




Corps	Taille (bits)	Année	Auteur(s)
$GF(p)$	431	2005	A. Joux et R. Lercier
$GF(p)$	530	2007	T. Kleinjung
$GF(p)$	768	2016	T. Kleinjung et al.
$GF(p)$	795	2019	F. Bourdot et al.
$GF(65537^{25})$	401	2005	A. Joux et R. Lercier
$GF(370801^{30})$	556	2005	A. Joux et R. Lercier
$GF(33553771^{47})$	1175	24-12-2012	A. Joux
$GF(33341353^{57})$	1425	06-01-2013	A. Joux
$GF(2^k)$	613	2005	A. Joux et R. Lercier
$GF(2^k)$	809	06-04-2013	Barbulescu et al.
$GF(2^k)$	1778	11-02-2013	A. Joux
$GF(2^k)$	1971	19-02-2013	R. Granger et al.
$GF(2^k)$	4080	22-03-2013	A. Joux
$GF(2^k)$	6120	11-04-2013	R. Granger et al.
$GF(2^k)$	6168	21-05-2013	A. Joux et al.
$GF(2^k)$	9324	31-01-2014	R. Granger et al.
$GF(3^k)$	676	2010	T. Hasashi et al.
$GF(3^k)$	923	2012	Kyushu, NICT and Fujitsu Labs
$GF(3^k)$	1551	27-01-2014	G. Adj et al.
$GF(3^k)$	3796	12-2014	A. Joux et al.
$GF(3^k)$	4841	18-07-2016	G. Adj et al.

# Équations de Lenstra (2004)

Année	Symétrique	Module (optimiste)	Module (conservateur)	Sous-groupe log discret
1989	61	515	649	122
1994	<b>64</b>	640	745	128
1999	68	781	850	136
2006	72	1007	1012	144
2018	<b>80</b>	1329	1478	160
<b>2019</b>	81	1358	1523	162
2025	85	(1538)	(1805)	(170)
2040	95	(2049)	(2644)	(190)
2066	112	(3154)	(4582)	(224)
2084	124	(4093)	(6318)	(248)
2090	<b>128</b>	(4440)	(6974)	(256)
2142	163	(8204)	(14423)	(326)
2282	<b>256</b>	(26268)	(53516)	512

- Coût de référence d'une attaque :  $40 \cdot 10^6$  [dollarday].
- Les nombres entre parenthèses doivent être pris avec **précaution**, étant donné que la date se situe loin dans le futur.
- Source : <http://www.keylength.com>

## keylength.com

Method	Date	Symmetric	Factoring Modulus	Discrete Key	Logarithm Group	Elliptic Curve	Hash
[1] Lenstra / Verheul 	2023	88	2054 1632	155	2054	166	175
[2] Lenstra Updated 	2023	84	1476 1708	167	1476	167	167
[3] ECRYPT	2018 - 2028	128	3072	256	3072	256	256
[4] NIST	2019 - 2030	112	2048	224	2048	224	224
[5] ANSSI	2021 - 2030	128	2048	200	2048	256	256
[6] NSA	-	256	3072	-	-	384	384
[7] RFC3766 	-	-	-	-	-	-	-
[8] BSI	2023 - 2026	128	3000	250	3000	250	256

- Il existe plein d'autres tables.
- A vous de choisir auxquelles vous faites confiance.



# ECRYPT

The goal of ECRYPT-CSA (Coordination & Support Action) is to strengthen European excellence in the area of cryptology. This report [3] on cryptographic algorithms, schemes, key sizes and protocols is a direct descendent of the reports produced by the ECRYPT I and II projects (2004-2012), and the ENISA reports (2013-2014). It provides rather conservative guiding principles, based on current state-of-the-art research, addressing construction of new systems with a long life cycle. This report is aimed to be a reference in the area, focusing on commercial online services that collect, store and process the data.

Protection	Symmetric	Factoring Modulus	Discrete Key	Logarithm Group	Elliptic Curve	Hash
Legacy standard level <i>Should not be used in new systems</i>	80	1024	160	1024	160	160
Near term protection <i>Security for at least ten years (2018-2028)</i>	128	3072	256	3072	256	256
Long-term protection <i>Security for thirty to fifty years (2018-2068)</i>	256	15360	512	15360	512	512

All key sizes are provided in bits. These are the minimal sizes for security.

**Click on a value to compare it with other methods.**

#### Recommended algorithms:

Block Ciphers: For near term use, AES-128 and for long term use, AES-256.

Hash Functions: For near term use, SHA-256 and for long term use, SHA-512 and SHA-3 with a 512-bit result.

Public Key Primitive: For near term use, 256-bit elliptic curves, and for long term use 512-bit elliptic curves.

#### Future algorithms (expected to remain secure in 10-50 year lifetime):

Block Ciphers: AES, Camellia, Serpent

Hash Functions: SHA2 (256, 384, 512, 512/256), SHA3 (256, 384, 512, SHAKE128, SHAKE256), Whirlpool-512, BLAKE (256, 584, 512)

Stream Ciphers: HC-128, Salsa20/20, ChaCha, SNOW 2.0, SNOW 3G, SOSEMANUK, Grain 128a



1. Choix de la Taille des Clefs
2. Théorème des Restes Chinois
3. Racines Carrées
4. Génération de Paramètres Cryptographiques

# Théorème des Restes Chinois

On souhaite résoudre le problème suivant, datant du troisième siècle de notre ère :

*Combien l'armée de Han Xing comporte-t-elle de soldats si, rangés par 3 colonnes, il reste deux soldats, rangés par 5 colonnes, il reste trois soldats et, rangés par 7 colonnes, il reste deux soldats ?*

Plus formellement, on souhaite résoudre le système d'équations suivant :

$$\begin{cases} x \equiv 2 & (\text{mod } 3) \\ x \equiv 3 & (\text{mod } 5) \\ x \equiv 2 & (\text{mod } 7) \end{cases}$$

# Théorème des Restes Chinois

## Théorème (Restes Chinois)

Soient  $m_1, m_2, \dots, m_n$  des entiers positifs **deux-à-deux premiers entre eux**, ainsi que  $a_1, a_2, \dots, a_n \in \mathbb{Z}$ . Une solution  $x$  du système

$$\begin{cases} x \equiv a_1 \pmod{m_1} \\ \dots \quad \dots \quad \dots \\ x \equiv a_n \pmod{m_n} \end{cases}$$

est donnée par

$$x \equiv \sum_{i=1}^n a_i \left( \left( \frac{m}{m_i} \right)^{-1} \pmod{m_i} \right) \frac{m}{m_i} \pmod{m}$$

où  $m = m_1 \cdots m_n$ .

Les solutions du système

$$\begin{cases} x \equiv 2 & (\text{mod } 3) \\ x \equiv 3 & (\text{mod } 5) \\ x \equiv 2 & (\text{mod } 7) \end{cases}$$

sont

$$2 \cdot (2 \cdot (5 \cdot 7)) + 3 \cdot (1 \cdot (3 \cdot 7)) + 2 \cdot (1 \cdot (3 \cdot 5)) \equiv 233 \equiv 23 \pmod{105}$$

# Théorème des Restes Chinois

- Pour l'instant, le théorème des reste chinois (CRT) ressemble à la formule magique suivante :

$$x \equiv \sum_{i=1}^n a_i \left( \left( \frac{m}{m_i} \right)^{-1} \pmod{m_i} \right) \frac{m}{m_i} \pmod{m}$$

- Le CRT est **beaucoup plus** que cela.

## Restes Chinois

Si  $m$  et  $n$  sont premiers entre eux, il existe un **isomorphisme d'anneau** entre  $\mathbb{Z}_m \times \mathbb{Z}_n$  et  $\mathbb{Z}_{mn}$ .

$$\mathbb{Z}_m \times \mathbb{Z}_n$$

- **Tuples** avec la première composante dans  $\mathbb{Z}_m$  et la deuxième dans  $\mathbb{Z}_n$ .
- Les calculs sont faits composante par composante.
- Il s'agit d'un **anneau** ! Les inverses sont calculés aussi composante par composante.
- Exemple dans  $\mathbb{Z}_3 \times \mathbb{Z}_5$  :  $(1, 3) + (2, 2) = (0, 0)$ .
- Exemple dans  $\mathbb{Z}_3 \times \mathbb{Z}_5$  :  $(1, 2) \cdot (2, 3) = (2, 1)$ .

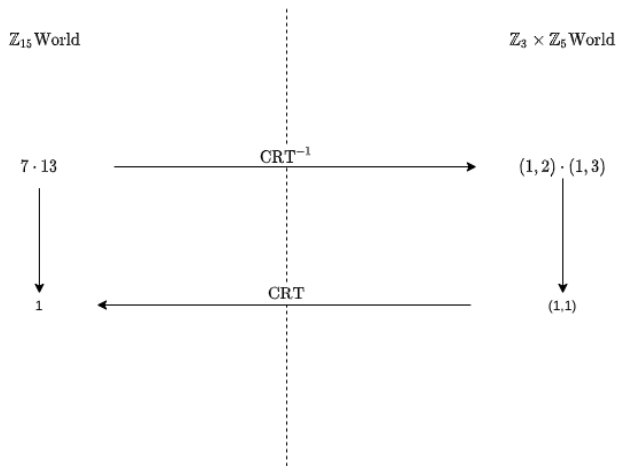
# Isomorphisme d'Anneau ?

Une fonction  $f$  est un **isomorphisme d'anneau** si

- $f : A \rightarrow B$ , pour deux anneaux  $A$  et  $B$ .
- $f$  est bijectif.
- Les opérations d'addition sont préservées :  
$$f(x + y) = f(x) + f(y).$$
- Les opérations de multiplication sont préservées :  
$$f(xy) = f(x)f(y).$$



## CRT (illustration)



## Retour au CRT

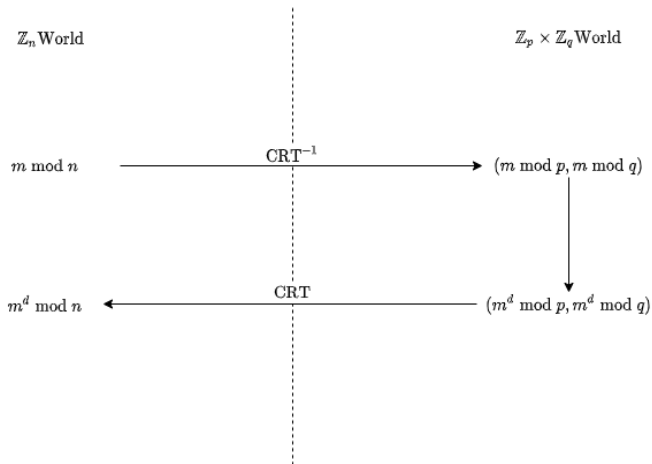
- On peut travailler arbitrairement dans  $\mathbb{Z}_{mn}$  ou dans  $\mathbb{Z}_m \times \mathbb{Z}_n$ . Les opérations sont **préservées**.
- Pour passer d'un élément  $x \in \mathbb{Z}_{mn}$  à  $(a, b) \in \mathbb{Z}_m \times \mathbb{Z}_n$ , on calcule  $a = x \bmod m$  et  $b = x \bmod n$ .
- Pour passer d'un élément  $(a, b) \in \mathbb{Z}_m \times \mathbb{Z}_n$  à  $x \in \mathbb{Z}_{mn}$ , on utilise la formule des restes chinois :

$$x \equiv (a(n^{-1} \bmod m)n + b(m^{-1} \bmod n)m) \pmod{mn}.$$

### Image

Les mondes  $\mathbb{Z}_{mn}$  et  $\mathbb{Z}_m \times \mathbb{Z}_n$  sont équivalents.

# Signatures RSA Rapides



- On peut accélérer les signatures RSA avec le **théorème des restes chinois**.

# Signatures RSA Rapides

- On peut accélérer les signatures RSA avec le **théorème des restes chinois**.
- Au lieu de calculer  $s = m^d \bmod n$ , on calcule  $s_p = m^{d \bmod p-1} \bmod p$  et  $s_q = m^{d \bmod q-1} \bmod q$ .
- On recombine ensuite le résultat avec le théorème des restes chinois :

$$s \equiv (s_p(q^{-1} \bmod p)q + s_q(p^{-1} \bmod q)p) \pmod{n}.$$

- Les deux exponentiations modulaires sont effectuées sur des paramètres possédant la **moitié de la taille** de  $n$ .
- **Réduction** du temps de calcul considérable.

1. Choix de la Taille des Clefs
2. Théorème des Restes Chinois
3. Racines Carrées
4. Génération de Paramètres Cryptographiques

## Question

Combien de racines carrées à le nombre 1 dans  $\mathbb{Z}_{15}$  ?

# Racines Carrées dans un Corps

- Dans un **corps**, un nombre a **au plus**  $n$  racines  $n$ ème.
- Dans  $\mathbb{Z}_p^*$ ,  $p > 2$  premier, un nombre a soit 0 soit 2 racines carrées.
- **Exemple** : dans  $\mathbb{Z}_7^*$ , 4 a deux racines carrées : 2 et 5.
- **Exemple** : dans  $\mathbb{Z}_7^*$ , 3 n'a pas de racines carrées.

# Racines Carrées dans un Anneau

- Dans  $\mathbb{Z}_{pq}^*$  avec,  $p, q > 2$  premiers et  $p \neq q$ , un nombre  $x$  a soit 0 soit 4 racines carrées.
- Cette propriété vient du **théorème des restes chinois** : si  $x$  a deux racines dans  $\mathbb{Z}_p$  et deux racines dans  $\mathbb{Z}_q$ , on a  $(\pm x_p)^2 = x \bmod p$  et  $(\pm x_q)^2 = x \bmod q$ . Soit  $f : \mathbb{Z}_p \times \mathbb{Z}_q \rightarrow \mathbb{Z}_{pq}$  l'isomorphisme du CRT, on a

$$f(x_p, x_q)f(x_p, x_q) = f(x_p x_p, x_q x_q) = f(x, x) = x$$

- $f(x_p, x_q)$  est donc une racine de  $x$  dans  $\mathbb{Z}_{pq}^*$ .
- Le même raisonnement tient pour  $f(\pm x_p, \pm x_q)$ .
- **Exemple** : 1 a 4 racines carrées dans  $\mathbb{Z}_{15}^*$  : 1, 4, 11, 14.



1. Choix de la Taille des Clefs
2. Théorème des Restes Chinois
3. Racines Carrées
4. Génération de Paramètres Cryptographiques

## Comment Générer des Nombres Premiers ?

1. Générer aléatoirement un nombre impair de taille voulue.
2. S'il n'est pas premier, retourner à l'étape 1.
3. S'il ne convient pas au cryptosystème retourner à l'étape 1.

La boucle est répétée environ  $\ln(p)$  fois, soit environ 900 fois en moyenne pour un nombre premier de 1300 bits.

# Test de Fermat

1. Choisir aléatoirement  $2 \leq a \leq n - 1$ .
2. Calculer  $x = a^{n-1} \bmod n$ . Si  $x \neq 1$ , alors  $n$  n'est pas premier.
3. Répéter suffisamment de fois les étapes 1 et 2. Si  $n$  passe tous les tests, alors il est peut-être premier (on dit que  $n$  est **pseudo-premier**).

## Problème

Il existe des nombres composés (non-premiers) passant presque toujours ce test ! Il s'agit des nombres de **Carmichael**, le plus petit étant  $561 = 3 \cdot 11 \cdot 17$ .

# Test de Miller-Rabin

- Version déterministe proposée par Gary L. Miller en 1976.  
Cette version repose sur l'hypothèse de Riemann généralisée, qui n'a jamais été démontrée.
- Version **probabiliste** décrite par Michael O. Rabin en 1980, qui ne repose sur aucune hypothèse non-démontrée.

# Test de Miller-Rabin

```

1: Ecrire  $n - 1 = 2^s d$  avec  $d$  impair.
2: for  $k$  itérations do
3:   Tirer un  $a \in [2, n - 1]$  aléatoire.
4:    $x \leftarrow a^d \bmod n, i \leftarrow 0$ 
5:   if  $x \neq 1$  then
6:     while  $x \neq n - 1$  do
7:        $x \leftarrow x^2 \bmod n$ 
8:        $i \leftarrow i + 1$ 
9:       if  $i = s$  then
10:        return Composite
11:      end if
12:      if  $x = 1$  then
13:        return Composite
14:      end if
15:    end while
16:  end if
17: end for
18: return Peut-être premier.

```

▷ Si  $x$  vaut 1, on ne peut rien dire

▷ Si  $i = s \Rightarrow a^{n-1} \neq 1$ .

▷ Si  $x = 1 \Rightarrow \sqrt{x} \neq \pm 1$ .

# Test de Miller-Rabin

## Résultat

La probabilité que le test de Miller-Rabin annonce  $n$  comme étant premier de manière erronée est inférieure à  $4^{-k}$ .

- Si  $a^{2^s d} \bmod n \neq 1$ , le nombre est composite (Test de Fermat).
- Sinon, on a forcément atteint 1 quelque part dans la chaîne de mises au carrés.
- L'équation  $x^2 \equiv 1 \pmod{n}$  possède au plus deux solutions si  $n$  est premier :  $\pm 1$  (corps).
- Le nombre précédant 1 était une racine de 1. Si cette racine diffère de  $-1$ ,  $\mathbb{Z}_n$  n'est pas un corps et  $n$  n'est pas premier.