

Introduction à la Cryptologie

Chapitre 6 : Le cryptosystème RSA

Michael Eisermann (Institut Fourier, UJF Grenoble)

Année 2008-2009
IF/IMAG, Master 1, S1-S2

document mis à jour le 7 juillet 2009



UNIVERSITE JOSEPH FOURIER
SCIENCES. TECHNOLOGIE. SANTÉ



www-fourier.ujf-grenoble.fr/~eiserm/cours#crypto

Développement mathématique : le cryptosystème RSA.

- Le petit théorème de Fermat.
- Le test de primalité selon Miller–Rabin.

Développement mathématique : le cryptosystème RSA.

- Le petit théorème de Fermat.
- Le test de primalité selon Miller–Rabin.

Développement algorithmique : implémentation de RSA.

- Assembler tous les algorithmes précédents.
- Production des grands nombres premiers.

- 1 Le cryptosystème RSA
- 2 Grands nombres premiers

- 1 Le cryptosystème RSA
 - Le petit théorème de Fermat
 - Le cryptosystème RSA
 - Signature et authentification
- 2 Grands nombres premiers

Le petit théorème de Fermat

Théorème (petit Fermat, formulation dans \mathbb{Z})

Si $p \in \mathbb{N}$ est premier alors tout $a \in \mathbb{Z}$ vérifie $a^p \equiv a \pmod{p}$.

Théorème (petit Fermat, formulation dans \mathbb{Z})

Si $p \in \mathbb{N}$ est premier alors tout $a \in \mathbb{Z}$ vérifie $a^p \equiv a \pmod{p}$.

Démonstration. Pour $a, b \in \mathbb{Z}$ on a

$$(a + b)^p = \sum_{k=0}^p \binom{p}{k} a^k b^{p-k} \equiv a^p + b^p \pmod{p}.$$

Théorème (petit Fermat, formulation dans \mathbb{Z})

Si $p \in \mathbb{N}$ est premier alors tout $a \in \mathbb{Z}$ vérifie $a^p \equiv a \pmod{p}$.

Démonstration. Pour $a, b \in \mathbb{Z}$ on a

$$(a + b)^p = \sum_{k=0}^p \binom{p}{k} a^k b^{p-k} \equiv a^p + b^p \pmod{p}.$$

Rappel : si $p \in \mathbb{N}$ est premier et $0 < k < p$, alors p divise $\binom{p}{k}$.

Théorème (petit Fermat, formulation dans \mathbb{Z})

Si $p \in \mathbb{N}$ est premier alors tout $a \in \mathbb{Z}$ vérifie $a^p \equiv a \pmod{p}$.

Démonstration. Pour $a, b \in \mathbb{Z}$ on a

$$(a + b)^p = \sum_{k=0}^p \binom{p}{k} a^k b^{p-k} \equiv a^p + b^p \pmod{p}.$$

Rappel : si $p \in \mathbb{N}$ est premier et $0 < k < p$, alors p divise $\binom{p}{k}$.

On a $0^p = 0$ et $1^p = 1$, puis on procède par récurrence sur $a \in \mathbb{N}$:

$$(a + 1)^p \equiv a^p + 1^p \equiv a + 1 \pmod{p}.$$

Théorème (petit Fermat, formulation dans \mathbb{Z})

Si $p \in \mathbb{N}$ est premier alors tout $a \in \mathbb{Z}$ vérifie $a^p \equiv a \pmod{p}$.

Démonstration. Pour $a, b \in \mathbb{Z}$ on a

$$(a + b)^p = \sum_{k=0}^p \binom{p}{k} a^k b^{p-k} \equiv a^p + b^p \pmod{p}.$$

Rappel : si $p \in \mathbb{N}$ est premier et $0 < k < p$, alors p divise $\binom{p}{k}$.

On a $0^p = 0$ et $1^p = 1$, puis on procède par récurrence sur $a \in \mathbb{N}$:

$$(a + 1)^p \equiv a^p + 1^p \equiv a + 1 \pmod{p}.$$

Si $a < 0$ on conclut par $a^p \equiv -(-a)^p \equiv -(-a) \equiv a \pmod{p}$. □

Théorème (petit Fermat, formulation dans \mathbb{Z})

Si $p \in \mathbb{N}$ est premier alors tout $a \in \mathbb{Z}$ vérifie $a^p \equiv a \pmod{p}$.

Démonstration. Pour $a, b \in \mathbb{Z}$ on a

$$(a + b)^p = \sum_{k=0}^p \binom{p}{k} a^k b^{p-k} \equiv a^p + b^p \pmod{p}.$$

Rappel : si $p \in \mathbb{N}$ est premier et $0 < k < p$, alors p divise $\binom{p}{k}$.

On a $0^p = 0$ et $1^p = 1$, puis on procède par récurrence sur $a \in \mathbb{N}$:

$$(a + 1)^p \equiv a^p + 1^p \equiv a + 1 \pmod{p}.$$

Si $a < 0$ on conclut par $a^p \equiv -(-a)^p \equiv -(-a) \equiv a \pmod{p}$. □

Corollaire (petit Fermat, formulation dans \mathbb{Z}/p)

Soit p premier. Tout $x \in \mathbb{Z}/p$ vérifie $x^p = x$ et tout $x \in \mathbb{Z}/p^*$ vérifie $x^{p-1} = \bar{1}$.

Le petit théorème de Fermat

Théorème (petit Fermat, formulation dans \mathbb{Z})

Si $p \in \mathbb{N}$ est premier alors tout $a \in \mathbb{Z}$ vérifie $a^p \equiv a \pmod{p}$.

Démonstration. Pour $a, b \in \mathbb{Z}$ on a

$$(a + b)^p = \sum_{k=0}^p \binom{p}{k} a^k b^{p-k} \equiv a^p + b^p \pmod{p}.$$

Rappel : si $p \in \mathbb{N}$ est premier et $0 < k < p$, alors p divise $\binom{p}{k}$.

On a $0^p = 0$ et $1^p = 1$, puis on procède par récurrence sur $a \in \mathbb{N}$:

$$(a + 1)^p \equiv a^p + 1^p \equiv a + 1 \pmod{p}.$$

Si $a < 0$ on conclut par $a^p \equiv -(-a)^p \equiv -(-a) \equiv a \pmod{p}$. □

Corollaire (petit Fermat, formulation dans \mathbb{Z}/p)

Soit p premier. Tout $x \in \mathbb{Z}/p$ vérifie $x^p = x$ et tout $x \in \mathbb{Z}/p^*$ vérifie $x^{p-1} = \bar{1}$.

Démonstration. Tout $x \in \mathbb{Z}/p$ s'écrit comme $x = \bar{a}$ où $a \in \mathbb{Z}$.

Théorème (petit Fermat, formulation dans \mathbb{Z})

Si $p \in \mathbb{N}$ est premier alors tout $a \in \mathbb{Z}$ vérifie $a^p \equiv a \pmod{p}$.

Démonstration. Pour $a, b \in \mathbb{Z}$ on a

$$(a + b)^p = \sum_{k=0}^p \binom{p}{k} a^k b^{p-k} \equiv a^p + b^p \pmod{p}.$$

Rappel : si $p \in \mathbb{N}$ est premier et $0 < k < p$, alors p divise $\binom{p}{k}$.

On a $0^p = 0$ et $1^p = 1$, puis on procède par récurrence sur $a \in \mathbb{N}$:

$$(a + 1)^p \equiv a^p + 1^p \equiv a + 1 \pmod{p}.$$

Si $a < 0$ on conclut par $a^p \equiv -(-a)^p \equiv -(-a) \equiv a \pmod{p}$. □

Corollaire (petit Fermat, formulation dans \mathbb{Z}/p)

Soit p premier. Tout $x \in \mathbb{Z}/p$ vérifie $x^p = x$ et tout $x \in \mathbb{Z}/p^*$ vérifie $x^{p-1} = \bar{1}$.

Démonstration. Tout $x \in \mathbb{Z}/p$ s'écrit comme $x = \bar{a}$ où $a \in \mathbb{Z}$.

Dans \mathbb{Z} on a $a^p \equiv a \pmod{p}$. Dans \mathbb{Z}/p on a donc $x^p = \overline{a^p} = \overline{a} = x$.

Le petit théorème de Fermat

Théorème (petit Fermat, formulation dans \mathbb{Z})

Si $p \in \mathbb{N}$ est premier alors tout $a \in \mathbb{Z}$ vérifie $a^p \equiv a \pmod{p}$.

Démonstration. Pour $a, b \in \mathbb{Z}$ on a

$$(a + b)^p = \sum_{k=0}^p \binom{p}{k} a^k b^{p-k} \equiv a^p + b^p \pmod{p}.$$

Rappel : si $p \in \mathbb{N}$ est premier et $0 < k < p$, alors p divise $\binom{p}{k}$.

On a $0^p = 0$ et $1^p = 1$, puis on procède par récurrence sur $a \in \mathbb{N}$:

$$(a + 1)^p \equiv a^p + 1^p \equiv a + 1 \pmod{p}.$$

Si $a < 0$ on conclut par $a^p \equiv -(-a)^p \equiv -(-a) \equiv a \pmod{p}$. □

Corollaire (petit Fermat, formulation dans \mathbb{Z}/p)

Soit p premier. Tout $x \in \mathbb{Z}/p$ vérifie $x^p = x$ et tout $x \in \mathbb{Z}/p^*$ vérifie $x^{p-1} = \bar{1}$.

Démonstration. Tout $x \in \mathbb{Z}/p$ s'écrit comme $x = \bar{a}$ où $a \in \mathbb{Z}$.

Dans \mathbb{Z} on a $a^p \equiv a \pmod{p}$. Dans \mathbb{Z}/p on a donc $x^p = \bar{a}^p = \overline{a^p} = \bar{a} = x$.

Pour $x \in \mathbb{Z}/p^* = \mathbb{Z}/p^\times$ l'égalité $x^p = x$ implique $x^{p-1} = \bar{1}$. □

L'observation suivante est un résultat d'Euler :

Corollaire

Soient $p, q \in \mathbb{N}$ deux nombres premiers distincts et $n = pq$.

L'observation suivante est un résultat d'Euler :

Corollaire

Soient $p, q \in \mathbb{N}$ deux nombres premiers distincts et $n = pq$.

Alors $\varphi(n) = (p - 1)(q - 1)$, et tout $x \in \mathbb{Z}/n^\times$ vérifie $x^{\varphi(n)} = \bar{1}$.

L'observation suivante est un résultat d'Euler :

Corollaire

Soient $p, q \in \mathbb{N}$ deux nombres premiers distincts et $n = pq$.

Alors $\varphi(n) = (p - 1)(q - 1)$, et tout $x \in \mathbb{Z}/n^\times$ vérifie $x^{\varphi(n)} = \bar{1}$.

Démonstration. On a $\Phi: \mathbb{Z}/n \xrightarrow{\sim} \mathbb{Z}/p \times \mathbb{Z}/q$ par le théorème chinois.

L'observation suivante est un résultat d'Euler :

Corollaire

Soient $p, q \in \mathbb{N}$ deux nombres premiers distincts et $n = pq$.

Alors $\varphi(n) = (p - 1)(q - 1)$, et tout $x \in \mathbb{Z}/n^\times$ vérifie $x^{\varphi(n)} = \bar{1}$.

Démonstration. On a $\Phi: \mathbb{Z}/n \xrightarrow{\sim} \mathbb{Z}/p \times \mathbb{Z}/q$ par le théorème chinois.

Si $\Phi(x) = (y, z)$, alors $x \in \mathbb{Z}/n^\times$ si et seulement si $y \in \mathbb{Z}/p^\times$ et $z \in \mathbb{Z}/q^\times$.

L'observation suivante est un résultat d'Euler :

Corollaire

Soient $p, q \in \mathbb{N}$ deux nombres premiers distincts et $n = pq$.

Alors $\varphi(n) = (p-1)(q-1)$, et tout $x \in \mathbb{Z}/n^\times$ vérifie $x^{\varphi(n)} = \bar{1}$.

Démonstration. On a $\Phi: \mathbb{Z}/n \xrightarrow{\sim} \mathbb{Z}/p \times \mathbb{Z}/q$ par le théorème chinois.

Si $\Phi(x) = (y, z)$, alors $x \in \mathbb{Z}/n^\times$ si et seulement si $y \in \mathbb{Z}/p^\times$ et $z \in \mathbb{Z}/q^\times$.

Dans ce cas on trouve

$$\begin{aligned}\Phi(x^{\varphi(n)}) &= \Phi(x)^{\varphi(n)} = (y, z)^{\varphi(n)} = (y^{\varphi(n)}, z^{\varphi(n)}) \\ &= ((y^{p-1})^{q-1}, (z^{q-1})^{p-1}) = (\bar{1}, \bar{1}) = \Phi(\bar{1}).\end{aligned}$$

L'observation suivante est un résultat d'Euler :

Corollaire

Soient $p, q \in \mathbb{N}$ deux nombres premiers distincts et $n = pq$.

Alors $\varphi(n) = (p-1)(q-1)$, et tout $x \in \mathbb{Z}/n^{\times}$ vérifie $x^{\varphi(n)} = \bar{1}$.

Démonstration. On a $\Phi: \mathbb{Z}/n \xrightarrow{\sim} \mathbb{Z}/p \times \mathbb{Z}/q$ par le théorème chinois.

Si $\Phi(x) = (y, z)$, alors $x \in \mathbb{Z}/n^{\times}$ si et seulement si $y \in \mathbb{Z}/p^{\times}$ et $z \in \mathbb{Z}/q^{\times}$.

Dans ce cas on trouve

$$\begin{aligned}\Phi(x^{\varphi(n)}) &= \Phi(x)^{\varphi(n)} = (y, z)^{\varphi(n)} = (y^{\varphi(n)}, z^{\varphi(n)}) \\ &= ((y^{p-1})^{q-1}, (z^{q-1})^{p-1}) = (\bar{1}, \bar{1}) = \Phi(\bar{1}).\end{aligned}$$

Comme Φ est bijective on conclut que $x^{\varphi(n)} = \bar{1}$.



Le petit théorème de Fermat : conséquences

L'observation suivante est à la base du cryptosystème RSA :

L'observation suivante est à la base du cryptosystème RSA :

Corollaire

Soient $p, q \in \mathbb{N}$ deux nombres premiers distincts et $n = pq$.

L'observation suivante est à la base du cryptosystème RSA :

Corollaire

Soient $p, q \in \mathbb{N}$ deux nombres premiers distincts et $n = pq$.

Soit $e = 1 + k \cdot \varphi(n)$ où $k \in \mathbb{N}$.

L'observation suivante est à la base du cryptosystème RSA :

Corollaire

Soient $p, q \in \mathbb{N}$ deux nombres premiers distincts et $n = pq$.

Soit $e = 1 + k \cdot \varphi(n)$ où $k \in \mathbb{N}$. Alors tout $x \in \mathbb{Z}/n$ vérifie $x^e = x$.

L'observation suivante est à la base du cryptosystème RSA :

Corollaire

Soient $p, q \in \mathbb{N}$ deux nombres premiers distincts et $n = pq$.

Soit $e = 1 + k \cdot \varphi(n)$ où $k \in \mathbb{N}$. Alors tout $x \in \mathbb{Z}/n$ vérifie $x^e = x$.

Démonstration. On a $\Phi: \mathbb{Z}/n \xrightarrow{\sim} \mathbb{Z}/p \times \mathbb{Z}/q$ par le théorème chinois.

L'observation suivante est à la base du cryptosystème RSA :

Corollaire

Soient $p, q \in \mathbb{N}$ deux nombres premiers distincts et $n = pq$.

Soit $e = 1 + k \cdot \varphi(n)$ où $k \in \mathbb{N}$. Alors tout $x \in \mathbb{Z}/n$ vérifie $x^e = x$.

Démonstration. On a $\Phi: \mathbb{Z}/n \xrightarrow{\sim} \mathbb{Z}/p \times \mathbb{Z}/q$ par le théorème chinois.

Pour $\Phi(x) = (y, z)$ on distingue quatre cas :

L'observation suivante est à la base du cryptosystème RSA :

Corollaire

Soient $p, q \in \mathbb{N}$ deux nombres premiers distincts et $n = pq$.

Soit $e = 1 + k \cdot \varphi(n)$ où $k \in \mathbb{N}$. Alors tout $x \in \mathbb{Z}/n$ vérifie $x^e = x$.

Démonstration. On a $\Phi: \mathbb{Z}/n \xrightarrow{\sim} \mathbb{Z}/p \times \mathbb{Z}/q$ par le théorème chinois.

Pour $\Phi(x) = (y, z)$ on distingue quatre cas :

- Si $y \neq \bar{0}$ et $z \neq \bar{0}$, alors

$$x^e = x^1 \cdot (x^{\varphi(n)})^k = x \cdot \bar{1} = x.$$

L'observation suivante est à la base du cryptosystème RSA :

Corollaire

Soient $p, q \in \mathbb{N}$ deux nombres premiers distincts et $n = pq$.

Soit $e = 1 + k \cdot \varphi(n)$ où $k \in \mathbb{N}$. Alors tout $x \in \mathbb{Z}/n$ vérifie $x^e = x$.

Démonstration. On a $\Phi: \mathbb{Z}/n \xrightarrow{\sim} \mathbb{Z}/p \times \mathbb{Z}/q$ par le théorème chinois.

Pour $\Phi(x) = (y, z)$ on distingue quatre cas :

- Si $y \neq \bar{0}$ et $z \neq \bar{0}$, alors

$$x^e = x^1 \cdot (x^{\varphi(n)})^k = x \cdot \bar{1} = x.$$

- Si $y = \bar{0}$ et $z \neq \bar{0}$, alors

$$\Phi(x^e) = (\bar{0}^e, z^e) = (\bar{0}, z^{1+k\varphi(n)}) = (\bar{0}, z \cdot (z^{q-1})^{k(p-1)}) = (\bar{0}, z) = \Phi(x).$$

L'observation suivante est à la base du cryptosystème RSA :

Corollaire

Soient $p, q \in \mathbb{N}$ deux nombres premiers distincts et $n = pq$.

Soit $e = 1 + k \cdot \varphi(n)$ où $k \in \mathbb{N}$. Alors tout $x \in \mathbb{Z}/n$ vérifie $x^e = x$.

Démonstration. On a $\Phi: \mathbb{Z}/n \xrightarrow{\sim} \mathbb{Z}/p \times \mathbb{Z}/q$ par le théorème chinois.

Pour $\Phi(x) = (y, z)$ on distingue quatre cas :

- Si $y \neq \bar{0}$ et $z \neq \bar{0}$, alors

$$x^e = x^1 \cdot (x^{\varphi(n)})^k = x \cdot \bar{1} = x.$$

- Si $y = \bar{0}$ et $z \neq \bar{0}$, alors

$$\Phi(x^e) = (\bar{0}^e, z^e) = (\bar{0}, z^{1+k\varphi(n)}) = (\bar{0}, z \cdot (z^{q-1})^{k(p-1)}) = (\bar{0}, z) = \Phi(x).$$

Comme Φ est bijective on conclut que $x^e = x$.

L'observation suivante est à la base du cryptosystème RSA :

Corollaire

Soient $p, q \in \mathbb{N}$ deux nombres premiers distincts et $n = pq$.

Soit $e = 1 + k \cdot \varphi(n)$ où $k \in \mathbb{N}$. Alors tout $x \in \mathbb{Z}/n$ vérifie $x^e = x$.

Démonstration. On a $\Phi: \mathbb{Z}/n \xrightarrow{\sim} \mathbb{Z}/p \times \mathbb{Z}/q$ par le théorème chinois.

Pour $\Phi(x) = (y, z)$ on distingue quatre cas :

- Si $y \neq \bar{0}$ et $z \neq \bar{0}$, alors

$$x^e = x^1 \cdot (x^{\varphi(n)})^k = x \cdot \bar{1} = x.$$

- Si $y = \bar{0}$ et $z \neq \bar{0}$, alors

$$\Phi(x^e) = (\bar{0}^e, z^e) = (\bar{0}, z^{1+k\varphi(n)}) = (\bar{0}, z \cdot (z^{q-1})^{k(p-1)}) = (\bar{0}, z) = \Phi(x).$$

Comme Φ est bijective on conclut que $x^e = x$.

- Si $y \neq \bar{0}$ et $z = \bar{0}$, l'argument est symétrique au précédent.

L'observation suivante est à la base du cryptosystème RSA :

Corollaire

Soient $p, q \in \mathbb{N}$ deux nombres premiers distincts et $n = pq$.

Soit $e = 1 + k \cdot \varphi(n)$ où $k \in \mathbb{N}$. Alors tout $x \in \mathbb{Z}/n$ vérifie $x^e = x$.

Démonstration. On a $\Phi: \mathbb{Z}/n \xrightarrow{\sim} \mathbb{Z}/p \times \mathbb{Z}/q$ par le théorème chinois.

Pour $\Phi(x) = (y, z)$ on distingue quatre cas :

- Si $y \neq \bar{0}$ et $z \neq \bar{0}$, alors

$$x^e = x^1 \cdot (x^{\varphi(n)})^k = x \cdot \bar{1} = x.$$

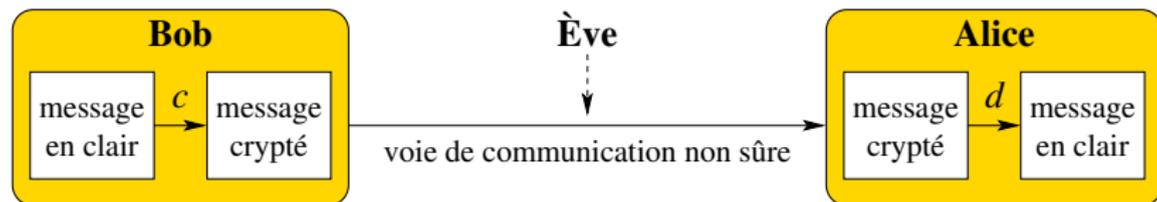
- Si $y = \bar{0}$ et $z \neq \bar{0}$, alors

$$\Phi(x^e) = (\bar{0}^e, z^e) = (\bar{0}, z^{1+k\varphi(n)}) = (\bar{0}, z \cdot (z^{q-1})^{k(p-1)}) = (\bar{0}, z) = \Phi(x).$$

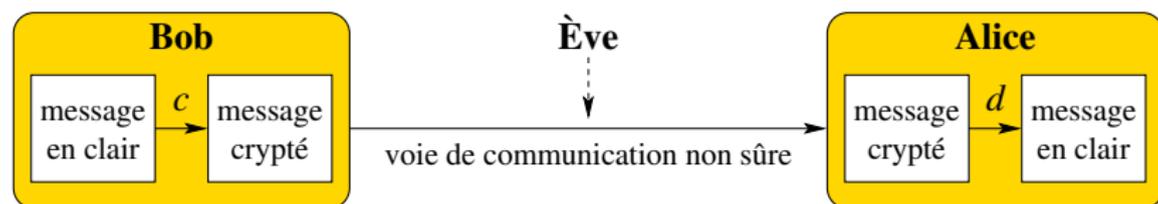
Comme Φ est bijective on conclut que $x^e = x$.

- Si $y \neq \bar{0}$ et $z = \bar{0}$, l'argument est symétrique au précédent.
- Si $y = \bar{0}$ et $z = \bar{0}$, alors $x = \bar{0}$ et $x^e = x = \bar{0}$. □

RSA : cryptage et décryptage

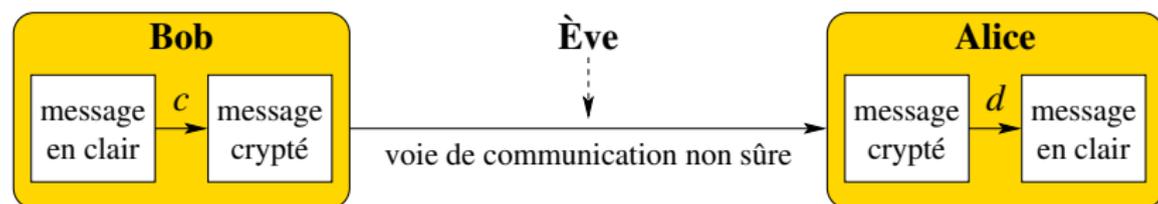


RSA : cryptage et décryptage



Clés : Soit $n = pq$ où $p, q \in \mathbb{N}$ sont deux nombres premiers distincts.

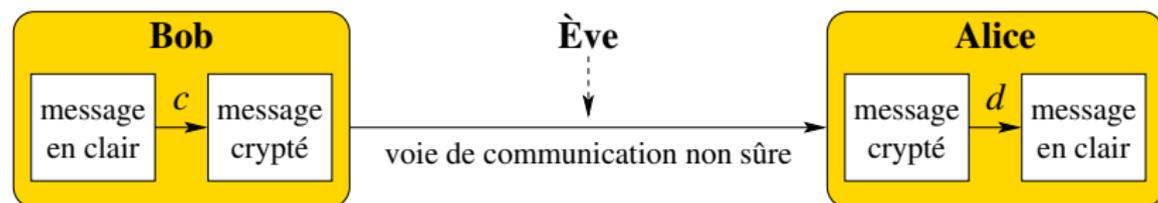
RSA : cryptage et décryptage



Clés : Soit $n = pq$ où $p, q \in \mathbb{N}$ sont deux nombres premiers distincts.

Soient $c, d \in \mathbb{N}$ tels que $cd \equiv 1 \text{ modulo } \varphi(n) = (p-1)(q-1)$.

RSA : cryptage et décryptage

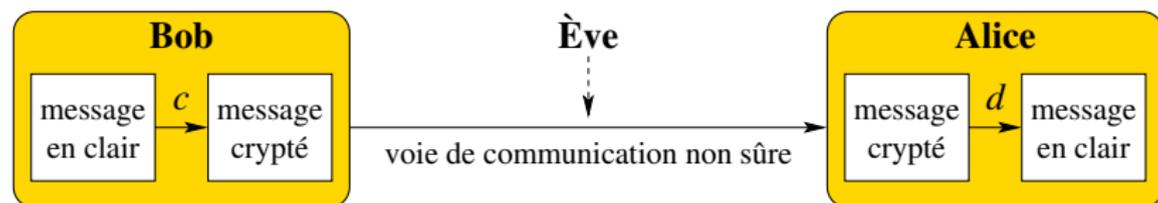


Clés : Soit $n = pq$ où $p, q \in \mathbb{N}$ sont deux nombres premiers distincts.

Soient $c, d \in \mathbb{N}$ tels que $cd \equiv 1$ modulo $\varphi(n) = (p-1)(q-1)$.

Cryptage : Bob représente son message par $m \in \mathbb{Z}/n$.

RSA : cryptage et décryptage



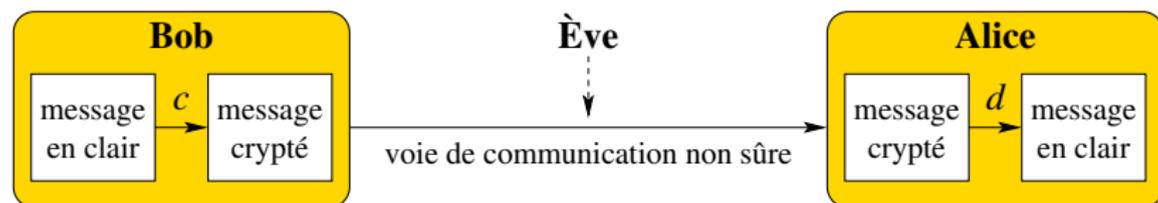
Clés : Soit $n = pq$ où $p, q \in \mathbb{N}$ sont deux nombres premiers distincts.

Soient $c, d \in \mathbb{N}$ tels que $cd \equiv 1$ modulo $\varphi(n) = (p-1)(q-1)$.

Cryptage : Bob représente son message par $m \in \mathbb{Z}/n$.

Il calcule le message crypté $\tilde{m} = m^c$.

RSA : cryptage et décryptage



Clés : Soit $n = pq$ où $p, q \in \mathbb{N}$ sont deux nombres premiers distincts.

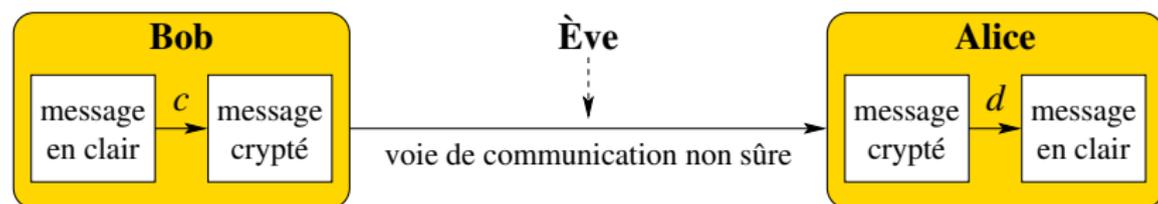
Soient $c, d \in \mathbb{N}$ tels que $cd \equiv 1$ modulo $\varphi(n) = (p-1)(q-1)$.

Cryptage : Bob représente son message par $m \in \mathbb{Z}/n$.

Il calcule le message crypté $\tilde{m} = m^c$.

Décryptage : Alice reçoit le message crypté $\tilde{m} \in \mathbb{Z}/n$.

RSA : cryptage et décryptage



Clés : Soit $n = pq$ où $p, q \in \mathbb{N}$ sont deux nombres premiers distincts.

Soient $c, d \in \mathbb{N}$ tels que $cd \equiv 1$ modulo $\varphi(n) = (p - 1)(q - 1)$.

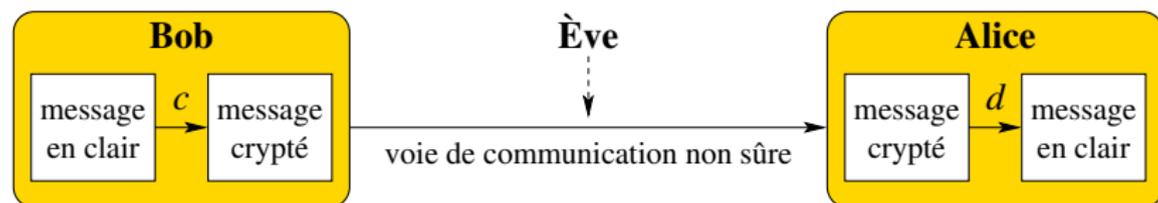
Cryptage : Bob représente son message par $m \in \mathbb{Z}/n$.

Il calcule le message crypté $\tilde{m} = m^c$.

Décryptage : Alice reçoit le message crypté $\tilde{m} \in \mathbb{Z}/n$.

Elle en déduit le message en clair $m = \tilde{m}^d$.

RSA : cryptage et décryptage



Clés : Soit $n = pq$ où $p, q \in \mathbb{N}$ sont deux nombres premiers distincts.

Soient $c, d \in \mathbb{N}$ tels que $cd \equiv 1$ modulo $\varphi(n) = (p - 1)(q - 1)$.

Cryptage : Bob représente son message par $m \in \mathbb{Z}/n$.

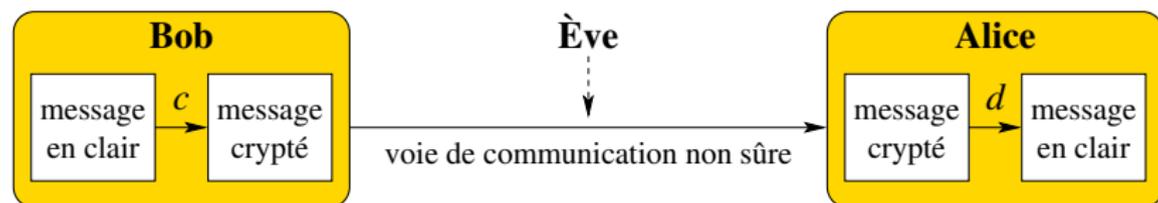
Il calcule le message crypté $\tilde{m} = m^c$.

Décryptage : Alice reçoit le message crypté $\tilde{m} \in \mathbb{Z}/n$.

Elle en déduit le message en clair $m = \tilde{m}^d$.

Pourquoi cela marche ?

RSA : cryptage et décryptage



Clés : Soit $n = pq$ où $p, q \in \mathbb{N}$ sont deux nombres premiers distincts.

Soient $c, d \in \mathbb{N}$ tels que $cd \equiv 1$ modulo $\varphi(n) = (p-1)(q-1)$.

Cryptage : Bob représente son message par $m \in \mathbb{Z}/n$.

Il calcule le message crypté $\tilde{m} = m^c$.

Décryptage : Alice reçoit le message crypté $\tilde{m} \in \mathbb{Z}/n$.

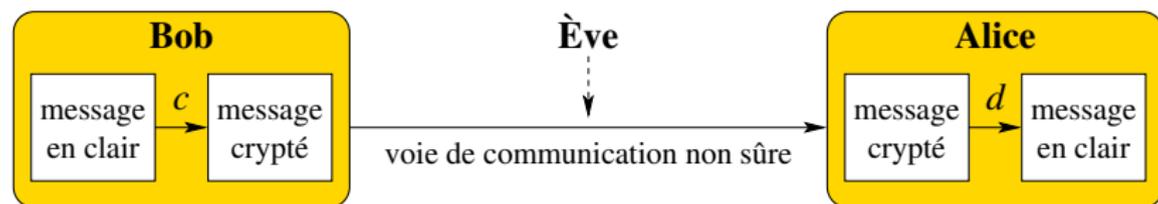
Elle en déduit le message en clair $m = \tilde{m}^d$.

Pourquoi cela marche ? On vient de montrer que

le cryptage $m \mapsto m^c$ et le décryptage $\tilde{m} \mapsto \tilde{m}^d$

définissent deux applications $\mathbb{Z}/n \leftrightarrow \mathbb{Z}/n$ qui sont mutuellement inverses.

RSA : cryptage et décryptage



Clés : Soit $n = pq$ où $p, q \in \mathbb{N}$ sont deux nombres premiers distincts.

Soient $c, d \in \mathbb{N}$ tels que $cd \equiv 1$ modulo $\varphi(n) = (p-1)(q-1)$.

Cryptage : Bob représente son message par $m \in \mathbb{Z}/n$.

Il calcule le message crypté $\tilde{m} = m^c$.

Décryptage : Alice reçoit le message crypté $\tilde{m} \in \mathbb{Z}/n$.

Elle en déduit le message en clair $m = \tilde{m}^d$.

Pourquoi cela marche ? On vient de montrer que

le cryptage $m \mapsto m^c$ et le décryptage $\tilde{m} \mapsto \tilde{m}^d$

définissent deux applications $\mathbb{Z}/n \leftrightarrow \mathbb{Z}/n$ qui sont mutuellement inverses.

◆ Ces calculs sont efficaces avec la puissance modulaire rapide !

Exercice

Pour $n = 5 \cdot 7$ et la clé de cryptage $c = 5$ déterminer la clé de décryptage d .
Crypter le message $m = \bar{4}$, puis décrypter le message ainsi crypté.

Exercice

Pour $n = 5 \cdot 7$ et la clé de cryptage $c = 5$ déterminer la clé de décryptage d .
Crypter le message $m = \bar{4}$, puis décrypter le message ainsi crypté.

Solution. On trouve $\varphi(n) = 4 \cdot 6 = 24$.

Exercice

Pour $n = 5 \cdot 7$ et la clé de cryptage $c = 5$ déterminer la clé de décryptage d .
Crypter le message $m = \bar{4}$, puis décrypter le message ainsi crypté.

Solution. On trouve $\varphi(n) = 4 \cdot 6 = 24$.

Pour $d = 5$ on a $cd = 25 \equiv 1 \pmod{24}$.

Exercice

Pour $n = 5 \cdot 7$ et la clé de cryptage $c = 5$ déterminer la clé de décryptage d .
Crypter le message $m = \bar{4}$, puis décrypter le message ainsi crypté.

Solution. On trouve $\varphi(n) = 4 \cdot 6 = 24$.

Pour $d = 5$ on a $cd = 25 \equiv 1 \pmod{24}$.

Dans $\mathbb{Z}/_{35}$ on crypte $m = \bar{4}$ en $\tilde{m} = m^c = \bar{4}^5 = \overline{1024} = \bar{9}$.

Exercice

Pour $n = 5 \cdot 7$ et la clé de cryptage $c = 5$ déterminer la clé de décryptage d .
Crypter le message $m = \bar{4}$, puis décrypter le message ainsi crypté.

Solution. On trouve $\varphi(n) = 4 \cdot 6 = 24$.

Pour $d = 5$ on a $cd = 25 \equiv 1 \pmod{24}$.

Dans $\mathbb{Z}/_{35}$ on crypte $m = \bar{4}$ en $\tilde{m} = m^c = \bar{4}^5 = \overline{1024} = \bar{9}$.

Puis on décrypte \tilde{m} en $\tilde{m}^d = \bar{9}^5 = \overline{59049} = \bar{4}$.



Exercice

Pour $n = 5 \cdot 7$ et la clé de cryptage $c = 5$ déterminer la clé de décryptage d .
Crypter le message $m = \bar{4}$, puis décrypter le message ainsi crypté.

Solution. On trouve $\varphi(n) = 4 \cdot 6 = 24$.

Pour $d = 5$ on a $cd = 25 \equiv 1 \pmod{24}$.

Dans $\mathbb{Z}/_{35}$ on crypte $m = \bar{4}$ en $\tilde{m} = m^c = \bar{4}^5 = \overline{1024} = \bar{9}$.

Puis on décrypte \tilde{m} en $\tilde{m}^d = \bar{9}^5 = \overline{59049} = \bar{4}$.



Exercice

Pour $n = 7 \cdot 11$ et la clé de cryptage $c = 7$ déterminer la clé de décryptage d .
Crypter le message $m = \bar{4}$, puis décrypter le message ainsi crypté.

Exercice

Pour $n = 5 \cdot 7$ et la clé de cryptage $c = 5$ déterminer la clé de décryptage d .
Crypter le message $m = \bar{4}$, puis décrypter le message ainsi crypté.

Solution. On trouve $\varphi(n) = 4 \cdot 6 = 24$.

Pour $d = 5$ on a $cd = 25 \equiv 1 \pmod{24}$.

Dans $\mathbb{Z}/_{35}$ on crypte $m = \bar{4}$ en $\tilde{m} = m^c = \bar{4}^5 = \overline{1024} = \bar{9}$.

Puis on décrypte \tilde{m} en $\tilde{m}^d = \bar{9}^5 = \overline{59049} = \bar{4}$.



Exercice

Pour $n = 7 \cdot 11$ et la clé de cryptage $c = 7$ déterminer la clé de décryptage d .
Crypter le message $m = \bar{4}$, puis décrypter le message ainsi crypté.

Solution. On trouve $\varphi(n) = 6 \cdot 10 = 60$.

Exercice

Pour $n = 5 \cdot 7$ et la clé de cryptage $c = 5$ déterminer la clé de décryptage d .
Crypter le message $m = \bar{4}$, puis décrypter le message ainsi crypté.

Solution. On trouve $\varphi(n) = 4 \cdot 6 = 24$.

Pour $d = 5$ on a $cd = 25 \equiv 1 \pmod{24}$.

Dans $\mathbb{Z}/_{35}$ on crypte $m = \bar{4}$ en $\tilde{m} = m^c = \bar{4}^5 = \overline{1024} = \bar{9}$.

Puis on décrypte \tilde{m} en $\tilde{m}^d = \bar{9}^5 = \overline{59049} = \bar{4}$.



Exercice

Pour $n = 7 \cdot 11$ et la clé de cryptage $c = 7$ déterminer la clé de décryptage d .
Crypter le message $m = \bar{4}$, puis décrypter le message ainsi crypté.

Solution. On trouve $\varphi(n) = 6 \cdot 10 = 60$.

Pour $d = 43$ on a $cd = 301 \equiv 1 \pmod{60}$.

Exercice

Pour $n = 5 \cdot 7$ et la clé de cryptage $c = 5$ déterminer la clé de décryptage d .
Crypter le message $m = \bar{4}$, puis décrypter le message ainsi crypté.

Solution. On trouve $\varphi(n) = 4 \cdot 6 = 24$.

Pour $d = 5$ on a $cd = 25 \equiv 1 \pmod{24}$.

Dans $\mathbb{Z}/_{35}$ on crypte $m = \bar{4}$ en $\tilde{m} = m^c = \bar{4}^5 = \overline{1024} = \bar{9}$.

Puis on décrypte \tilde{m} en $\tilde{m}^d = \bar{9}^5 = \overline{59049} = \bar{4}$.



Exercice

Pour $n = 7 \cdot 11$ et la clé de cryptage $c = 7$ déterminer la clé de décryptage d .
Crypter le message $m = \bar{4}$, puis décrypter le message ainsi crypté.

Solution. On trouve $\varphi(n) = 6 \cdot 10 = 60$.

Pour $d = 43$ on a $cd = 301 \equiv 1 \pmod{60}$.

Dans $\mathbb{Z}/_{77}$ on crypte $m = \bar{4}$ en $\tilde{m} = m^c = \bar{4}^7 = \overline{16384} = \bar{60}$.

Exercice

Pour $n = 5 \cdot 7$ et la clé de cryptage $c = 5$ déterminer la clé de décryptage d .
Crypter le message $m = \bar{4}$, puis décrypter le message ainsi crypté.

Solution. On trouve $\varphi(n) = 4 \cdot 6 = 24$.

Pour $d = 5$ on a $cd = 25 \equiv 1 \pmod{24}$.

Dans $\mathbb{Z}/_{35}$ on crypte $m = \bar{4}$ en $\tilde{m} = m^c = \bar{4}^5 = \overline{1024} = \bar{9}$.

Puis on décrypte \tilde{m} en $\tilde{m}^d = \bar{9}^5 = \overline{59049} = \bar{4}$.



Exercice

Pour $n = 7 \cdot 11$ et la clé de cryptage $c = 7$ déterminer la clé de décryptage d .
Crypter le message $m = \bar{4}$, puis décrypter le message ainsi crypté.

Solution. On trouve $\varphi(n) = 6 \cdot 10 = 60$.

Pour $d = 43$ on a $cd = 301 \equiv 1 \pmod{60}$.

Dans $\mathbb{Z}/_{77}$ on crypte $m = \bar{4}$ en $\tilde{m} = m^c = \bar{4}^7 = \overline{16384} = \bar{60}$.

Puis on décrypte \tilde{m} en $\tilde{m}^d = \overline{60}^{43} = \bar{4}$.

(Utiliser une calculatrice et la puissance modulaire.)



RSA : production des clés

Alice construit un triplet (n, c, d) vérifiant $cd \equiv 1 \text{ modulo } \varphi(n)$:

RSA : production des clés

Alice construit un triplet (n, c, d) vérifiant $cd \equiv 1$ modulo $\varphi(n)$:

- 1 Elle choisit deux grands nombres premiers distincts $p, q \in \mathbb{N}$.

RSA : production des clés

Alice construit un triplet (n, c, d) vérifiant $cd \equiv 1$ modulo $\varphi(n)$:

- 1 Elle choisit deux grands nombres premiers distincts $p, q \in \mathbb{N}$.
- 2 Elle calcule leur produit $n = pq$ ainsi que $\varphi(n) = (p - 1)(q - 1)$.

RSA : production des clés

Alice construit un triplet (n, c, d) vérifiant $cd \equiv 1$ modulo $\varphi(n)$:

- 1 Elle choisit deux grands nombres premiers distincts $p, q \in \mathbb{N}$.
- 2 Elle calcule leur produit $n = pq$ ainsi que $\varphi(n) = (p - 1)(q - 1)$.
- 3 Elle choisit un entier $c \in \mathbb{N}$ premier avec $\varphi(n)$.

RSA : production des clés

Alice construit un triplet (n, c, d) vérifiant $cd \equiv 1$ modulo $\varphi(n)$:

- 1 Elle choisit deux grands nombres premiers distincts $p, q \in \mathbb{N}$.
- 2 Elle calcule leur produit $n = pq$ ainsi que $\varphi(n) = (p - 1)(q - 1)$.
- 3 Elle choisit un entier $c \in \mathbb{N}$ premier avec $\varphi(n)$.
- 4 Elle calcule un inverse $d \in \mathbb{N}$ tel que $cd \equiv 1$ modulo $\varphi(n)$.

RSA : production des clés

Alice construit un triplet (n, c, d) vérifiant $cd \equiv 1$ modulo $\varphi(n)$:

- 1 Elle choisit deux grands nombres premiers distincts $p, q \in \mathbb{N}$.
- 2 Elle calcule leur produit $n = pq$ ainsi que $\varphi(n) = (p - 1)(q - 1)$.
- 3 Elle choisit un entier $c \in \mathbb{N}$ premier avec $\varphi(n)$.
- 4 Elle calcule un inverse $d \in \mathbb{N}$ tel que $cd \equiv 1$ modulo $\varphi(n)$.

Tous ces calculs sont efficaces ; il reste à préciser l'étape (1).

RSA : production des clés

Alice construit un triplet (n, c, d) vérifiant $cd \equiv 1$ modulo $\varphi(n)$:

- 1 Elle choisit deux grands nombres premiers distincts $p, q \in \mathbb{N}$.
- 2 Elle calcule leur produit $n = pq$ ainsi que $\varphi(n) = (p - 1)(q - 1)$.
- 3 Elle choisit un entier $c \in \mathbb{N}$ premier avec $\varphi(n)$.
- 4 Elle calcule un inverse $d \in \mathbb{N}$ tel que $cd \equiv 1$ modulo $\varphi(n)$.

Tous ces calculs sont efficaces ; il reste à préciser l'étape (1).

Clé publique : La clé publique d'Alice est la paire (n, c) .

RSA : production des clés

Alice construit un triplet (n, c, d) vérifiant $cd \equiv 1$ modulo $\varphi(n)$:

- 1 Elle choisit deux grands nombres premiers distincts $p, q \in \mathbb{N}$.
- 2 Elle calcule leur produit $n = pq$ ainsi que $\varphi(n) = (p - 1)(q - 1)$.
- 3 Elle choisit un entier $c \in \mathbb{N}$ premier avec $\varphi(n)$.
- 4 Elle calcule un inverse $d \in \mathbb{N}$ tel que $cd \equiv 1$ modulo $\varphi(n)$.

Tous ces calculs sont efficaces ; il reste à préciser l'étape (1).

Clé publique : La clé publique d'Alice est la paire (n, c) .

Ainsi tout le monde peut envoyer des messages cryptés $\tilde{m} = m^c$ à Alice.

RSA : production des clés

Alice construit un triplet (n, c, d) vérifiant $cd \equiv 1$ modulo $\varphi(n)$:

- 1 Elle choisit deux grands nombres premiers distincts $p, q \in \mathbb{N}$.
- 2 Elle calcule leur produit $n = pq$ ainsi que $\varphi(n) = (p - 1)(q - 1)$.
- 3 Elle choisit un entier $c \in \mathbb{N}$ premier avec $\varphi(n)$.
- 4 Elle calcule un inverse $d \in \mathbb{N}$ tel que $cd \equiv 1$ modulo $\varphi(n)$.

Tous ces calculs sont efficaces ; il reste à préciser l'étape (1).

Clé publique : La clé publique d'Alice est la paire (n, c) .

Ainsi tout le monde peut envoyer des messages cryptés $\tilde{m} = m^c$ à Alice.

Clé secrète : La clé secrète d'Alice est d .

RSA : production des clés

Alice construit un triplet (n, c, d) vérifiant $cd \equiv 1$ modulo $\varphi(n)$:

- 1 Elle choisit deux grands nombres premiers distincts $p, q \in \mathbb{N}$.
- 2 Elle calcule leur produit $n = pq$ ainsi que $\varphi(n) = (p - 1)(q - 1)$.
- 3 Elle choisit un entier $c \in \mathbb{N}$ premier avec $\varphi(n)$.
- 4 Elle calcule un inverse $d \in \mathbb{N}$ tel que $cd \equiv 1$ modulo $\varphi(n)$.

Tous ces calculs sont efficaces ; il reste à préciser l'étape (1).

Clé publique : La clé publique d'Alice est la paire (n, c) .

Ainsi tout le monde peut envoyer des messages cryptés $\tilde{m} = m^c$ à Alice.

Clé secrète : La clé secrète d'Alice est d .

Ainsi Alice sait décrypter des messages en calculant $m = \tilde{m}^d$.

RSA : production des clés

Alice construit un triplet (n, c, d) vérifiant $cd \equiv 1$ modulo $\varphi(n)$:

- 1 Elle choisit deux grands nombres premiers distincts $p, q \in \mathbb{N}$.
- 2 Elle calcule leur produit $n = pq$ ainsi que $\varphi(n) = (p - 1)(q - 1)$.
- 3 Elle choisit un entier $c \in \mathbb{N}$ premier avec $\varphi(n)$.
- 4 Elle calcule un inverse $d \in \mathbb{N}$ tel que $cd \equiv 1$ modulo $\varphi(n)$.

Tous ces calculs sont efficaces ; il reste à préciser l'étape (1).

Clé publique : La clé publique d'Alice est la paire (n, c) .

Ainsi tout le monde peut envoyer des messages cryptés $\tilde{m} = m^c$ à Alice.

Clé secrète : La clé secrète d'Alice est d .

Ainsi Alice sait décrypter des messages en calculant $m = \tilde{m}^d$.

Hypothèses : La sécurité de RSA repose sur certaines hypothèses :

RSA : production des clés

Alice construit un triplet (n, c, d) vérifiant $cd \equiv 1 \text{ modulo } \varphi(n)$:

- 1 Elle choisit deux grands nombres premiers distincts $p, q \in \mathbb{N}$.
- 2 Elle calcule leur produit $n = pq$ ainsi que $\varphi(n) = (p - 1)(q - 1)$.
- 3 Elle choisit un entier $c \in \mathbb{N}$ premier avec $\varphi(n)$.
- 4 Elle calcule un inverse $d \in \mathbb{N}$ tel que $cd \equiv 1 \text{ modulo } \varphi(n)$.

Tous ces calculs sont efficaces ; il reste à préciser l'étape (1).

Clé publique : La clé publique d'Alice est la paire (n, c) .

Ainsi tout le monde peut envoyer des messages cryptés $\tilde{m} = m^c$ à Alice.

Clé secrète : La clé secrète d'Alice est d .

Ainsi Alice sait décrypter des messages en calculant $m = \tilde{m}^d$.

Hypothèses : La sécurité de RSA repose sur certaines hypothèses :

- Il est difficile de retrouver les facteurs p et q à partir de n .

RSA : production des clés

Alice construit un triplet (n, c, d) vérifiant $cd \equiv 1 \text{ modulo } \varphi(n)$:

- 1 Elle choisit deux grands nombres premiers distincts $p, q \in \mathbb{N}$.
- 2 Elle calcule leur produit $n = pq$ ainsi que $\varphi(n) = (p - 1)(q - 1)$.
- 3 Elle choisit un entier $c \in \mathbb{N}$ premier avec $\varphi(n)$.
- 4 Elle calcule un inverse $d \in \mathbb{N}$ tel que $cd \equiv 1 \text{ modulo } \varphi(n)$.

Tous ces calculs sont efficaces ; il reste à préciser l'étape (1).

Clé publique : La clé publique d'Alice est la paire (n, c) .

Ainsi tout le monde peut envoyer des messages cryptés $\tilde{m} = m^c$ à Alice.

Clé secrète : La clé secrète d'Alice est d .

Ainsi Alice sait décrypter des messages en calculant $m = \tilde{m}^d$.

Hypothèses : La sécurité de RSA repose sur certaines hypothèses :

- Il est difficile de retrouver les facteurs p et q à partir de n .
- Sans connaître p et q , il est difficile de trouver $\varphi(n)$ à partir de n .

RSA : production des clés

Alice construit un triplet (n, c, d) vérifiant $cd \equiv 1 \text{ modulo } \varphi(n)$:

- 1 Elle choisit deux grands nombres premiers distincts $p, q \in \mathbb{N}$.
- 2 Elle calcule leur produit $n = pq$ ainsi que $\varphi(n) = (p - 1)(q - 1)$.
- 3 Elle choisit un entier $c \in \mathbb{N}$ premier avec $\varphi(n)$.
- 4 Elle calcule un inverse $d \in \mathbb{N}$ tel que $cd \equiv 1 \text{ modulo } \varphi(n)$.

Tous ces calculs sont efficaces ; il reste à préciser l'étape (1).

Clé publique : La clé publique d'Alice est la paire (n, c) .

Ainsi tout le monde peut envoyer des messages cryptés $\tilde{m} = m^c$ à Alice.

Clé secrète : La clé secrète d'Alice est d .

Ainsi Alice sait décrypter des messages en calculant $m = \tilde{m}^d$.

Hypothèses : La sécurité de RSA repose sur certaines hypothèses :

- Il est difficile de retrouver les facteurs p et q à partir de n .
- Sans connaître p et q , il est difficile de trouver $\varphi(n)$ à partir de n .
- Sans connaître $\varphi(n)$, il est difficile de trouver d à partir de (n, c) .

RSA : production des clés

Alice construit un triplet (n, c, d) vérifiant $cd \equiv 1 \text{ modulo } \varphi(n)$:

- 1 Elle choisit deux grands nombres premiers distincts $p, q \in \mathbb{N}$.
- 2 Elle calcule leur produit $n = pq$ ainsi que $\varphi(n) = (p - 1)(q - 1)$.
- 3 Elle choisit un entier $c \in \mathbb{N}$ premier avec $\varphi(n)$.
- 4 Elle calcule un inverse $d \in \mathbb{N}$ tel que $cd \equiv 1 \text{ modulo } \varphi(n)$.

Tous ces calculs sont efficaces ; il reste à préciser l'étape (1).

Clé publique : La clé publique d'Alice est la paire (n, c) .

Ainsi tout le monde peut envoyer des messages cryptés $\tilde{m} = m^c$ à Alice.

Clé secrète : La clé secrète d'Alice est d .

Ainsi Alice sait décrypter des messages en calculant $m = \tilde{m}^d$.

Hypothèses : La sécurité de RSA repose sur certaines hypothèses :

- Il est difficile de retrouver les facteurs p et q à partir de n .
- Sans connaître p et q , il est difficile de trouver $\varphi(n)$ à partir de n .
- Sans connaître $\varphi(n)$, il est difficile de trouver d à partir de (n, c) .
- Sans connaître d , il est difficile de trouver m à partir de \tilde{m} .

Règles évidentes de bon sens :

Règles évidentes de bon sens :

- Il faut choisir p et q suffisamment grands.
Actuellement on utilise entre 1024 et 2048 bits.

Règles évidentes de bon sens :

- Il faut choisir p et q suffisamment grands.
Actuellement on utilise entre 1024 et 2048 bits.
- Il faut changer la clé (n, c, d) régulièrement.
Ceci limite le temps pour des attaques à brute force.

Règles évidentes de bon sens :

- Il faut choisir p et q suffisamment grands.
Actuellement on utilise entre 1024 et 2048 bits.
- Il faut changer la clé (n, c, d) régulièrement.
Ceci limite le temps pour des attaques à brute force.
- Transmettre les clés publiques par un « serveur sûr ».
Sinon aucun cryptosystème ne peut fonctionner.

RSA : problèmes et attaques connus

Règles évidentes de bon sens :

- Il faut choisir p et q suffisamment grands.
Actuellement on utilise entre 1024 et 2048 bits.
- Il faut changer la clé (n, c, d) régulièrement.
Ceci limite le temps pour des attaques à brute force.
- Transmettre les clés publiques par un « serveur sûr ».
Sinon aucun cryptosystème ne peut fonctionner.

Failles de sécurité plus subtiles :

Règles évidentes de bon sens :

- Il faut choisir p et q suffisamment grands.
Actuellement on utilise entre 1024 et 2048 bits.
- Il faut changer la clé (n, c, d) régulièrement.
Ceci limite le temps pour des attaques à brute force.
- Transmettre les clés publiques par un « serveur sûr ».
Sinon aucun cryptosystème ne peut fonctionner.

Failles de sécurité plus subtiles :

- La factorisation est rapide pour certains produits $n = pq$.
Éviter des premiers « faibles » et choisir des premiers « forts ».

Règles évidentes de bon sens :

- Il faut choisir p et q suffisamment grands.
Actuellement on utilise entre 1024 et 2048 bits.
- Il faut changer la clé (n, c, d) régulièrement.
Ceci limite le temps pour des attaques à brute force.
- Transmettre les clés publiques par un « serveur sûr ».
Sinon aucun cryptosystème ne peut fonctionner.

Faibles de sécurité plus subtiles :

- La factorisation est rapide pour certains produits $n = pq$.
Éviter des premiers « faibles » et choisir des premiers « forts ».
- Certaines clés publiques c révèlent de l'information sur d .
Éviter des clés « faibles » et choisir des clés « fortes ».

Règles évidentes de bon sens :

- Il faut choisir p et q suffisamment grands.
Actuellement on utilise entre 1024 et 2048 bits.
- Il faut changer la clé (n, c, d) régulièrement.
Ceci limite le temps pour des attaques à brute force.
- Transmettre les clés publiques par un « serveur sûr ».
Sinon aucun cryptosystème ne peut fonctionner.

Failles de sécurité plus subtiles :

- La factorisation est rapide pour certains produits $n = pq$.
Éviter des premiers « faibles » et choisir des premiers « forts ».
- Certaines clés publiques c révèlent de l'information sur d .
Éviter des clés « faibles » et choisir des clés « fortes ».
- Certains messages \tilde{m} révèlent de l'information sur m ou d .
Bien choisir le codage des messages $m \in \mathbb{Z}/n$.

Règles évidentes de bon sens :

- Il faut choisir p et q suffisamment grands.
Actuellement on utilise entre 1024 et 2048 bits.
- Il faut changer la clé (n, c, d) régulièrement.
Ceci limite le temps pour des attaques à brute force.
- Transmettre les clés publiques par un « serveur sûr ».
Sinon aucun cryptosystème ne peut fonctionner.

Failles de sécurité plus subtiles :

- La factorisation est rapide pour certains produits $n = pq$.
Éviter des premiers « faibles » et choisir des premiers « forts ».
- Certaines clés publiques c révèlent de l'information sur d .
Éviter des clés « faibles » et choisir des clés « fortes ».
- Certains messages \tilde{m} révèlent de l'information sur m ou d .
Bien choisir le codage des messages $m \in \mathbb{Z}/n$.



Les détails dépendent de l'état de l'art de la cryptanalyse !

Règles évidentes de bon sens :

- Il faut choisir p et q suffisamment grands.
Actuellement on utilise entre 1024 et 2048 bits.
- Il faut changer la clé (n, c, d) régulièrement.
Ceci limite le temps pour des attaques à brute force.
- Transmettre les clés publiques par un « serveur sûr ».
Sinon aucun cryptosystème ne peut fonctionner.

Failles de sécurité plus subtiles :

- La factorisation est rapide pour certains produits $n = pq$.
Éviter des premiers « faibles » et choisir des premiers « forts ».
- Certaines clés publiques c révèlent de l'information sur d .
Éviter des clés « faibles » et choisir des clés « fortes ».
- Certains messages \tilde{m} révèlent de l'information sur m ou d .
Bien choisir le codage des messages $m \in \mathbb{Z}/n$.

 Les détails dépendent de l'état de l'art de la cryptanalyse !

La sécurité du cryptosystème RSA n'est pas mathématiquement prouvée.

Règles évidentes de bon sens :

- Il faut choisir p et q suffisamment grands.
Actuellement on utilise entre 1024 et 2048 bits.
- Il faut changer la clé (n, c, d) régulièrement.
Ceci limite le temps pour des attaques à brute force.
- Transmettre les clés publiques par un « serveur sûr ».
Sinon aucun cryptosystème ne peut fonctionner.

Faibles de sécurité plus subtiles :

- La factorisation est rapide pour certains produits $n = pq$.
Éviter des premiers « faibles » et choisir des premiers « forts ».
- Certaines clés publiques c révèlent de l'information sur d .
Éviter des clés « faibles » et choisir des clés « fortes ».
- Certains messages \tilde{m} révèlent de l'information sur m ou d .
Bien choisir le codage des messages $m \in \mathbb{Z}/n$.

 Les détails dépendent de l'état de l'art de la cryptanalyse !
La sécurité du cryptosystème RSA n'est pas mathématiquement prouvée.
Elle n'est pas absolue mais relative aux connaissances actuelles.

Objectif : Alice veut *signer* un document électronique, de sorte que tout le monde puisse *vérifier* la signature, mais personne ne puisse la *falsifier*.

Objectif : Alice veut *signer* un document électronique, de sorte que tout le monde puisse *vérifier* la signature, mais personne ne puisse la *falsifier*.

Signature : Alice signe son message m à l'aide d'une fonction de hachage

$$h: \{\text{messages}\} \rightarrow \mathbb{Z}/n, \quad m \mapsto h(m).$$

Objectif : Alice veut *signer* un document électronique, de sorte que tout le monde puisse *vérifier* la signature, mais personne ne puisse la *falsifier*.

Signature : Alice signe son message m à l'aide d'une fonction de hachage

$$h: \{\text{messages}\} \rightarrow \mathbb{Z}/n, \quad m \mapsto h(m).$$

C'est une *somme de contrôle* ou une *empreinte cryptographique*.

Objectif : Alice veut *signer* un document électronique, de sorte que tout le monde puisse *vérifier* la signature, mais personne ne puisse la *falsifier*.

Signature : Alice signe son message m à l'aide d'une fonction de hachage

$$h: \{\text{messages}\} \rightarrow \mathbb{Z}/n, \quad m \mapsto h(m).$$

C'est une *somme de contrôle* ou une *empreinte cryptographique*.

Alice calcule $s = h(m)^d$ dans \mathbb{Z}/n . Le message signé est (m, s) .

Objectif : Alice veut *signer* un document électronique, de sorte que tout le monde puisse *vérifier* la signature, mais personne ne puisse la *falsifier*.

Signature : Alice signe son message m à l'aide d'une fonction de hachage

$$h: \{\text{messages}\} \rightarrow \mathbb{Z}/n, \quad m \mapsto h(m).$$

C'est une *somme de contrôle* ou une *empreinte cryptographique*.

Alice calcule $s = h(m)^d$ dans \mathbb{Z}/n . Le message signé est (m, s) .

Authentification : Bob reçoit (m, s) et récupère la clé publique (n, c) d'Alice.

Objectif : Alice veut *signer* un document électronique, de sorte que tout le monde puisse *vérifier* la signature, mais personne ne puisse la *falsifier*.

Signature : Alice signe son message m à l'aide d'une fonction de hachage

$$h: \{\text{messages}\} \rightarrow \mathbb{Z}/n, \quad m \mapsto h(m).$$

C'est une *somme de contrôle* ou une *empreinte cryptographique*.

Alice calcule $s = h(m)^d$ dans \mathbb{Z}/n . Le message signé est (m, s) .

Authentification : Bob reçoit (m, s) et récupère la clé publique (n, c) d'Alice.

Il calcule s^c dans \mathbb{Z}/n et le compare à l'empreinte $h(m)$.

Objectif : Alice veut *signer* un document électronique, de sorte que tout le monde puisse *vérifier* la signature, mais personne ne puisse la *falsifier*.

Signature : Alice signe son message m à l'aide d'une fonction de hachage

$$h: \{\text{messages}\} \rightarrow \mathbb{Z}/n, \quad m \mapsto h(m).$$

C'est une *somme de contrôle* ou une *empreinte cryptographique*.

Alice calcule $s = h(m)^d$ dans \mathbb{Z}/n . Le message signé est (m, s) .

Authentification : Bob reçoit (m, s) et récupère la clé publique (n, c) d'Alice.

Il calcule s^c dans \mathbb{Z}/n et le compare à l'empreinte $h(m)$.

Si $s^c = h(m)$ alors le message est authentifié.

Objectif : Alice veut *signer* un document électronique, de sorte que tout le monde puisse *vérifier* la signature, mais personne ne puisse la *falsifier*.

Signature : Alice signe son message m à l'aide d'une fonction de hachage

$$h: \{\text{messages}\} \rightarrow \mathbb{Z}/n, \quad m \mapsto h(m).$$

C'est une *somme de contrôle* ou une *empreinte cryptographique*.

Alice calcule $s = h(m)^d$ dans \mathbb{Z}/n . Le message signé est (m, s) .

Authentification : Bob reçoit (m, s) et récupère la clé publique (n, c) d'Alice.

Il calcule s^c dans \mathbb{Z}/n et le compare à l'empreinte $h(m)$.

Si $s^c = h(m)$ alors le message est authentifié.

Hypothèses : La sécurité repose sur les mêmes hypothèses qu'avant.

Objectif : Alice veut *signer* un document électronique, de sorte que tout le monde puisse *vérifier* la signature, mais personne ne puisse la *falsifier*.

Signature : Alice signe son message m à l'aide d'une fonction de hachage

$$h: \{\text{messages}\} \rightarrow \mathbb{Z}/n, \quad m \mapsto h(m).$$

C'est une *somme de contrôle* ou une *empreinte cryptographique*.

Alice calcule $s = h(m)^d$ dans \mathbb{Z}/n . Le message signé est (m, s) .

Authentification : Bob reçoit (m, s) et récupère la clé publique (n, c) d'Alice.

Il calcule s^c dans \mathbb{Z}/n et le compare à l'empreinte $h(m)$.

Si $s^c = h(m)$ alors le message est authentifié.

Hypothèses : La sécurité repose sur les mêmes hypothèses qu'avant.

En plus, il faut une fonction de hachage fiable dans le sens que pour le message m il est difficile de construire m' tel que $h(m) = h(m')$.

1 Le cryptosystème RSA

2 Grands nombres premiers

- Le théorème des nombres premiers
- Critères de primalité : de Fermat à Miller–Rabin
- Production de grands nombres premiers aléatoires

Théorème (Euclide, environ 300 avant notre ère)

Il existe une infinité de nombres premiers.



Théorème (Euclide, environ 300 avant notre ère)

Il existe une infinité de nombres premiers. □

Algorithme 6.2 crible d'Ératosthène (environ 275–194 avant notre ère)

Entrée: la liste $(p_0 = 2, p_1 = 3, \dots, p_{k-1})$ des k plus petits nombres premiers

Sortie: la liste $(p_0 = 2, p_1 = 3, \dots, p_{k-1}, p_k)$ des $k + 1$ plus petits nombres premiers

$p \leftarrow p_{k-1} + 2, \quad i \leftarrow 1$

tant que $p_i^2 \leq p$ **faire si** $p_i \nmid p$ **alors** $i \leftarrow i + 1$ **sinon** $p \leftarrow p + 2, \quad i \leftarrow 1$

retourner la liste prolongée $(p_0 = 2, p_1 = 3, \dots, p_{k-1}, p)$

Théorème (Euclide, environ 300 avant notre ère)

Il existe une infinité de nombres premiers. □

Algorithme 6.3 crible d'Ératosthène (environ 275–194 avant notre ère)

Entrée: la liste $(p_0 = 2, p_1 = 3, \dots, p_{k-1})$ des k plus petits nombres premiers

Sortie: la liste $(p_0 = 2, p_1 = 3, \dots, p_{k-1}, p_k)$ des $k + 1$ plus petits nombres premiers

$p \leftarrow p_{k-1} + 2, \quad i \leftarrow 1$

tant que $p_i^2 \leq p$ **faire si** $p_i \nmid p$ **alors** $i \leftarrow i + 1$ **sinon** $p \leftarrow p + 2, \quad i \leftarrow 1$

retourner la liste prolongée $(p_0 = 2, p_1 = 3, \dots, p_{k-1}, p)$

Pour $n \in \mathbb{N}$ on note $\pi(n)$ le nombre des entiers premiers dans $\{1, \dots, n\}$.

Théorème (Euclide, environ 300 avant notre ère)

Il existe une infinité de nombres premiers. □

Algorithme 6.4 crible d'Ératosthène (environ 275–194 avant notre ère)

Entrée: la liste $(p_0 = 2, p_1 = 3, \dots, p_{k-1})$ des k plus petits nombres premiers

Sortie: la liste $(p_0 = 2, p_1 = 3, \dots, p_{k-1}, p_k)$ des $k + 1$ plus petits nombres premiers

$p \leftarrow p_{k-1} + 2, \quad i \leftarrow 1$

tant que $p_i^2 \leq p$ **faire si** $p_i \nmid p$ **alors** $i \leftarrow i + 1$ **sinon** $p \leftarrow p + 2, \quad i \leftarrow 1$

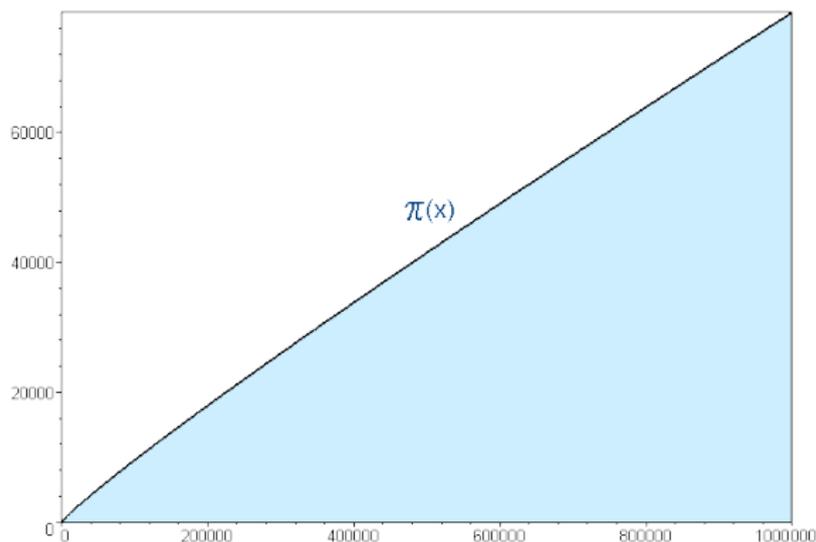
retourner la liste prolongée $(p_0 = 2, p_1 = 3, \dots, p_{k-1}, p)$

Pour $n \in \mathbb{N}$ on note $\pi(n)$ le nombre des entiers premiers dans $\{1, \dots, n\}$.

k	$\pi(10^k)$	k	$\pi(10^k)$	k	$\pi(10^k)$
1	4	8	5 761 455	15	29 844 570 422 669
2	25	9	50 847 534	16	279 238 341 033 925
3	168	10	455 052 511	17	2 623 557 157 654 233
4	1 229	11	4 118 054 813	18	24 739 954 287 740 860
5	9 592	12	37 607 912 018	19	234 057 667 276 344 607
6	78 498	13	346 065 536 839	20	2 220 819 602 560 918 840
7	664 579	14	3 204 941 750 802	21	21 127 269 486 018 731 928

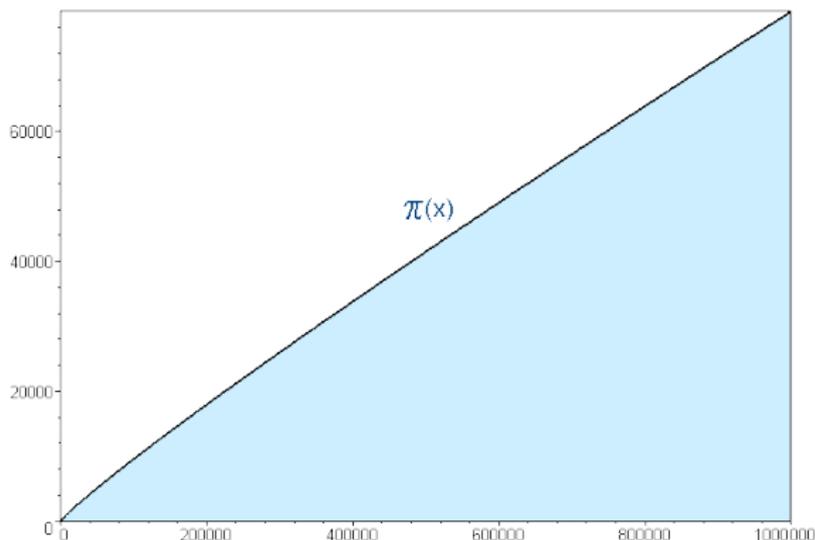
Le théorème des nombres premiers : asymptotique

En contemplant ce tableau jusqu'à 10^6 Gauss conjectura que $\pi(n) \sim n/\ln n$.



Le théorème des nombres premiers : asymptotique

En contemplant ce tableau jusqu'à 10^6 Gauss conjectura que $\pi(n) \sim n / \ln n$.



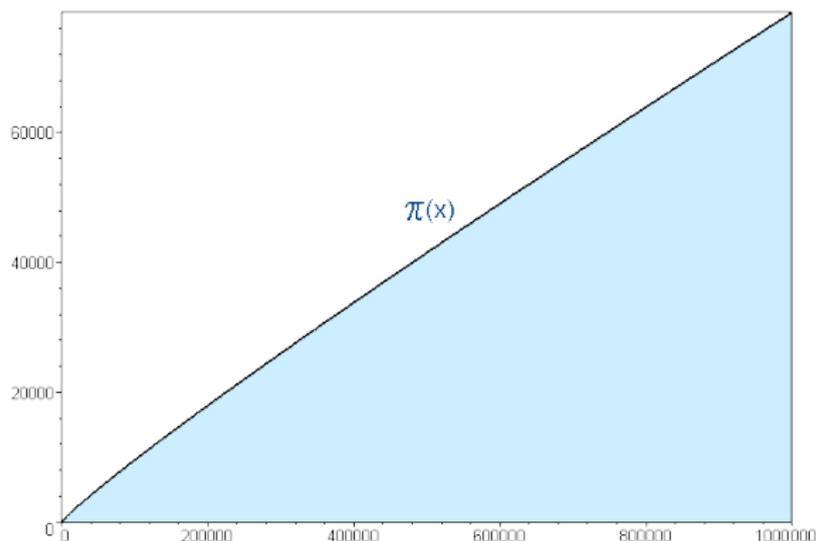
Théorème (Hadamard et de la Vallée Poussin 1896)

On a l'équivalence asymptotique $\pi(n) \sim n / \ln n$.



Le théorème des nombres premiers : asymptotique

En contemplant ce tableau jusqu'à 10^6 Gauss conjectura que $\pi(n) \sim n / \ln n$.



Théorème (Hadamard et de la Vallée Poussin 1896)

On a l'équivalence asymptotique $\pi(n) \sim n / \ln n$. □

Ceci veut dire que le quotient $\frac{\pi(n)}{n / \ln n}$ converge vers 1 pour $n \rightarrow \infty$.

La version quantitative suivante donne un encadrement explicite :

La version quantitative suivante donne un encadrement explicite :

Théorème (Rosser–Schoenfeld 1962)

Pour $n \geq 59$ on a l'encadrement

$$\frac{n}{\ln n} \left(1 + \frac{1}{2 \ln n} \right) < \pi(n) < \frac{n}{\ln n} \left(1 + \frac{3}{2 \ln n} \right).$$

La version quantitative suivante donne un encadrement explicite :

Théorème (Rosser–Schoenfeld 1962)

Pour $n \geq 59$ on a l'encadrement

$$\frac{n}{\ln n} \left(1 + \frac{1}{2 \ln n} \right) < \pi(n) < \frac{n}{\ln n} \left(1 + \frac{3}{2 \ln n} \right).$$

Corollaire

L'intervalle $\{a \in \mathbb{Z} \mid n \leq a < 2n\}$ contient $\sim n / \ln n$ nombres premiers. \square

La version quantitative suivante donne un encadrement explicite :

Théorème (Rosser–Schoenfeld 1962)

Pour $n \geq 59$ on a l'encadrement

$$\frac{n}{\ln n} \left(1 + \frac{1}{2 \ln n} \right) < \pi(n) < \frac{n}{\ln n} \left(1 + \frac{3}{2 \ln n} \right).$$

Corollaire

L'intervalle $\{a \in \mathbb{Z} \mid n \leq a < 2n\}$ contient $\sim n / \ln n$ nombres premiers. \square

Corollaire

Si l'on tire au hasard un entier dans l'intervalle $\{n, \dots, 2n\}$
alors la probabilité de tomber sur un nombre premier est $\sim 1 / \ln n$. \square

Critère

Un entier $n \in \mathbb{N}$ est premier si et seulement si $\mathbb{Z}/n^{\times} = \mathbb{Z}/n^$.*

Critère

Un entier $n \in \mathbb{N}$ est premier si et seulement si $\mathbb{Z}/n^{\times} = \mathbb{Z}/n^{}$.*

Il est composé si et seulement s'il existe $0 < a < n$ tel que $\text{pgcd}(a, n) > 1$.

Critère

Un entier $n \in \mathbb{N}$ est premier si et seulement si $\mathbb{Z}/n^{\times} = \mathbb{Z}/n^{}$.*

Il est composé si et seulement s'il existe $0 < a < n$ tel que $\text{pgcd}(a, n) > 1$.

Test naïf de primalité :

Critère

Un entier $n \in \mathbb{N}$ est premier si et seulement si $\mathbb{Z}/n^{\times} = \mathbb{Z}/n^{}$.*

Il est composé si et seulement s'il existe $0 < a < n$ tel que $\text{pgcd}(a, n) > 1$.

Test naïf de primalité :

On tire a au hasard dans $\{1, \dots, n - 1\}$.

Critère

Un entier $n \in \mathbb{N}$ est premier si et seulement si $\mathbb{Z}/n^{\times} = \mathbb{Z}/n^{}$.*

Il est composé si et seulement s'il existe $0 < a < n$ tel que $\text{pgcd}(a, n) > 1$.

Test naïf de primalité :

On tire a au hasard dans $\{1, \dots, n-1\}$.

Si $\text{pgcd}(a, n) > 1$ alors n est composé (et on connaît un facteur).

Critère

Un entier $n \in \mathbb{N}$ est premier si et seulement si $\mathbb{Z}/n^{\times} = \mathbb{Z}/n^{}$.*

Il est composé si et seulement s'il existe $0 < a < n$ tel que $\text{pgcd}(a, n) > 1$.

Test naïf de primalité :

On tire a au hasard dans $\{1, \dots, n-1\}$.

Si $\text{pgcd}(a, n) > 1$ alors n est composé (et on connaît un facteur).

Pourquoi cela ne marche pas bien ?

Critère

Un entier $n \in \mathbb{N}$ est premier si et seulement si $\mathbb{Z}/n^{\times} = \mathbb{Z}/n^{}$.*

Il est composé si et seulement s'il existe $0 < a < n$ tel que $\text{pgcd}(a, n) > 1$.

Test naïf de primalité :

On tire a au hasard dans $\{1, \dots, n-1\}$.

Si $\text{pgcd}(a, n) > 1$ alors n est composé (et on connaît un facteur).

Pourquoi cela ne marche pas bien ?

Supposons que n est composé, $n = p_1^{e_1} \cdots p_k^{e_k}$.

Critère

Un entier $n \in \mathbb{N}$ est premier si et seulement si $\mathbb{Z}/n^{\times} = \mathbb{Z}/n^{}$.*

Il est composé si et seulement s'il existe $0 < a < n$ tel que $\text{pgcd}(a, n) > 1$.

Test naïf de primalité :

On tire a au hasard dans $\{1, \dots, n-1\}$.

Si $\text{pgcd}(a, n) > 1$ alors n est composé (et on connaît un facteur).

Pourquoi cela ne marche pas bien ?

Supposons que n est composé, $n = p_1^{e_1} \cdots p_k^{e_k}$.

La probabilité qu'un élément $x \in \mathbb{Z}/n$ choisi au hasard soit inversible vaut

$$\frac{\varphi(n)}{n} = \prod_{i=1}^k \left(1 - \frac{1}{p_i}\right).$$

Critère

Un entier $n \in \mathbb{N}$ est premier si et seulement si $\mathbb{Z}/n^{\times} = \mathbb{Z}/n^{}$.*

Il est composé si et seulement s'il existe $0 < a < n$ tel que $\text{pgcd}(a, n) > 1$.

Test naïf de primalité :

On tire a au hasard dans $\{1, \dots, n-1\}$.

Si $\text{pgcd}(a, n) > 1$ alors n est composé (et on connaît un facteur).

Pourquoi cela ne marche pas bien ?

Supposons que n est composé, $n = p_1^{e_1} \cdots p_k^{e_k}$.

La probabilité qu'un élément $x \in \mathbb{Z}/n$ choisi au hasard soit inversible vaut

$$\frac{\varphi(n)}{n} = \prod_{i=1}^k \left(1 - \frac{1}{p_i}\right).$$

Si n n'a pas de petits facteurs, alors cette probabilité est proche de 1.

Critère

Un entier $n \in \mathbb{N}$ est premier si et seulement si $\mathbb{Z}/n^{\times} = \mathbb{Z}/n^{}$.*

Il est composé si et seulement s'il existe $0 < a < n$ tel que $\text{pgcd}(a, n) > 1$.

Test naïf de primalité :

On tire a au hasard dans $\{1, \dots, n-1\}$.

Si $\text{pgcd}(a, n) > 1$ alors n est composé (et on connaît un facteur).

Pourquoi cela ne marche pas bien ?

Supposons que n est composé, $n = p_1^{e_1} \cdots p_k^{e_k}$.

La probabilité qu'un élément $x \in \mathbb{Z}/n$ choisi au hasard soit inversible vaut

$$\frac{\varphi(n)}{n} = \prod_{i=1}^k \left(1 - \frac{1}{p_i}\right).$$

Si n n'a pas de petits facteurs, alors cette probabilité est proche de 1.

La probabilité de tomber sur un élément non inversible est proche de 0.



Critère (petit théorème de Fermat)

Si n est premier, alors tout $x \in \mathbb{Z}/n^$ vérifie $x^{n-1} = 1$.*

Critère (petit théorème de Fermat)

Si n est premier, alors tout $x \in \mathbb{Z}/_n^$ vérifie $x^{n-1} = 1$.*

Test de primalité selon Fermat :

Critère (petit théorème de Fermat)

Si n est premier, alors tout $x \in \mathbb{Z}/n^$ vérifie $x^{n-1} = 1$.*

Test de primalité selon Fermat :

On tire x au hasard dans \mathbb{Z}/n^* puis on vérifie si $x^{n-1} = 1$.

Critère (petit théorème de Fermat)

Si n est premier, alors tout $x \in \mathbb{Z}/n^$ vérifie $x^{n-1} = 1$.*

Test de primalité selon Fermat :

On tire x au hasard dans \mathbb{Z}/n^* puis on vérifie si $x^{n-1} = 1$.

Si $x^{n-1} \neq 1$, alors n est composé (sans que l'on connaisse de facteur).

Critère (petit théorème de Fermat)

Si n est premier, alors tout $x \in \mathbb{Z}/n^$ vérifie $x^{n-1} = 1$.*

Test de primalité selon Fermat :

On tire x au hasard dans \mathbb{Z}/n^* puis on vérifie si $x^{n-1} = 1$.

Si $x^{n-1} \neq 1$, alors n est composé (sans que l'on connaisse de facteur).

Dans ce cas on dit que x est un *témoin de décomposabilité* de n ,
ou encore que x *témoigne* contre la primalité de n .

Critère (petit théorème de Fermat)

Si n est premier, alors tout $x \in \mathbb{Z}/_n^*$ vérifie $x^{n-1} = 1$.

Test de primalité selon Fermat :

On tire x au hasard dans $\mathbb{Z}/_n^*$ puis on vérifie si $x^{n-1} = 1$.

Si $x^{n-1} \neq 1$, alors n est composé (sans que l'on connaisse de facteur).

Dans ce cas on dit que x est un *témoin de décomposabilité* de n ,
ou encore que x *témoigne* contre la primalité de n .

Est-ce efficace ? Parfois oui, mais pas toujours :

Observation (Korsel 1899, Carmichael 1910)

Il existe des nombres composés n tel que $a^n \equiv a \pmod n$ pour tout $a \in \mathbb{Z}$.

Critère (petit théorème de Fermat)

Si n est premier, alors tout $x \in \mathbb{Z}/n^*$ vérifie $x^{n-1} = 1$.

Test de primalité selon Fermat :

On tire x au hasard dans \mathbb{Z}/n^* puis on vérifie si $x^{n-1} = 1$.

Si $x^{n-1} \neq 1$, alors n est composé (sans que l'on connaisse de facteur).

Dans ce cas on dit que x est un *témoin de décomposabilité* de n ,
ou encore que x *témoigne* contre la primalité de n .

Est-ce efficace ? Parfois oui, mais pas toujours :

Observation (Korsel 1899, Carmichael 1910)

Il existe des nombres composés n tel que $a^n \equiv a \pmod n$ pour tout $a \in \mathbb{Z}$.

Les plus petits exemples sont 561, 1105, 1729, 2465, 2821, 6601, 8911, 10585, 15841, 29341, 41041, ... On sait depuis 1994 qu'il en existe une infinité.

Critère (petit théorème de Fermat)

Si n est premier, alors tout $x \in \mathbb{Z}/n^*$ vérifie $x^{n-1} = 1$.

Test de primalité selon Fermat :

On tire x au hasard dans \mathbb{Z}/n^* puis on vérifie si $x^{n-1} = 1$.

Si $x^{n-1} \neq 1$, alors n est composé (sans que l'on connaisse de facteur).

Dans ce cas on dit que x est un *témoin de décomposabilité* de n ,
ou encore que x *témoigne* contre la primalité de n .

Est-ce efficace ? Parfois oui, mais pas toujours :

Observation (Korsel 1899, Carmichael 1910)

Il existe des nombres composés n tel que $a^n \equiv a \pmod n$ pour tout $a \in \mathbb{Z}$.

Les plus petits exemples sont 561, 1105, 1729, 2465, 2821, 6601, 8911, 10585, 15841, 29341, 41041, ... On sait depuis 1994 qu'il en existe une infinité.

Dans ces cas tout élément inversible $x \in \mathbb{Z}/n^\times$ vérifie $x^{n-1} = 1$.

Critère (petit théorème de Fermat)

Si n est premier, alors tout $x \in \mathbb{Z}/n^*$ vérifie $x^{n-1} = 1$.

Test de primalité selon Fermat :

On tire x au hasard dans \mathbb{Z}/n^* puis on vérifie si $x^{n-1} = 1$.

Si $x^{n-1} \neq 1$, alors n est composé (sans que l'on connaisse de facteur).

Dans ce cas on dit que x est un *témoin de décomposabilité* de n ,
ou encore que x *témoigne* contre la primalité de n .

Est-ce efficace ? Parfois oui, mais pas toujours :

Observation (Korsel 1899, Carmichael 1910)

Il existe des nombres composés n tel que $a^n \equiv a \pmod n$ pour tout $a \in \mathbb{Z}$.

Les plus petits exemples sont 561, 1105, 1729, 2465, 2821, 6601, 8911, 10585, 15841, 29341, 41041, ... On sait depuis 1994 qu'il en existe une infinité.

Dans ces cas tout élément inversible $x \in \mathbb{Z}/n^\times$ vérifie $x^{n-1} = 1$.

Ainsi trouver un témoin est aussi rare que dans le test naïf précédent.



L'astuce de la racine carrée

Si n est premier, alors tout $x \in \mathbb{Z}/n^*$ vérifie $x^{n-1} = 1$.

L'astuce de la racine carrée

Si n est premier, alors tout $x \in \mathbb{Z}/n^*$ vérifie $x^{n-1} = 1$.

Or, ce beau critère n'est pas infaillible. Essayons de l'optimiser :

L'astuce de la racine carrée

Si n est premier, alors tout $x \in \mathbb{Z}/n^*$ vérifie $x^{n-1} = 1$.

Or, ce beau critère n'est pas infaillible. Essayons de l'optimiser :

Critère

Si n est un nombre premier impair, alors tout $x \in \mathbb{Z}/n^$ vérifie $x^{\frac{n-1}{2}} = \pm 1$. \square*

L'astuce de la racine carrée

Si n est premier, alors tout $x \in \mathbb{Z}/n^*$ vérifie $x^{n-1} = 1$.

Or, ce beau critère n'est pas infaillible. Essayons de l'optimiser :

Critère

Si n est un nombre premier impair, alors tout $x \in \mathbb{Z}/n^$ vérifie $x^{\frac{n-1}{2}} = \pm 1$. \square*

Démonstration. D'après Fermat $y = x^{\frac{n-1}{2}}$ vérifie $y^2 = 1$.

L'astuce de la racine carrée

Si n est premier, alors tout $x \in \mathbb{Z}/n^*$ vérifie $x^{n-1} = 1$.

Or, ce beau critère n'est pas infallible. Essayons de l'optimiser :

Critère

Si n est un nombre premier impair, alors tout $x \in \mathbb{Z}/n^$ vérifie $x^{\frac{n-1}{2}} = \pm 1$. \square*

Démonstration. D'après Fermat $y = x^{\frac{n-1}{2}}$ vérifie $y^2 = 1$.

Si n est premier, alors $y^2 = 1$ n'admet que deux solutions dans \mathbb{Z}/n :

L'astuce de la racine carrée

Si n est premier, alors tout $x \in \mathbb{Z}/n^*$ vérifie $x^{n-1} = 1$.

Or, ce beau critère n'est pas infallible. Essayons de l'optimiser :

Critère

Si n est un nombre premier impair, alors tout $x \in \mathbb{Z}/n^$ vérifie $x^{\frac{n-1}{2}} = \pm 1$. \square*

Démonstration. D'après Fermat $y = x^{\frac{n-1}{2}}$ vérifie $y^2 = 1$.

Si n est premier, alors $y^2 = 1$ n'admet que deux solutions dans \mathbb{Z}/n :

$$y^2 = 1$$

$$\Leftrightarrow y^2 - 1 = 0$$

$$\Leftrightarrow (y - 1)(y + 1) = 0$$

$$\Leftrightarrow y = 1 \text{ ou } y = -1.$$

L'astuce de la racine carrée

Si n est premier, alors tout $x \in \mathbb{Z}/n^*$ vérifie $x^{n-1} = 1$.

Or, ce beau critère n'est pas infallible. Essayons de l'optimiser :

Critère

Si n est un nombre premier impair, alors tout $x \in \mathbb{Z}/n^$ vérifie $x^{\frac{n-1}{2}} = \pm 1$. \square*

Démonstration. D'après Fermat $y = x^{\frac{n-1}{2}}$ vérifie $y^2 = 1$.

Si n est premier, alors $y^2 = 1$ n'admet que deux solutions dans \mathbb{Z}/n :

$$y^2 = 1$$

$$\Leftrightarrow y^2 - 1 = 0$$

$$\Leftrightarrow (y - 1)(y + 1) = 0$$

$$\Leftrightarrow y = 1 \text{ ou } y = -1.$$

Pour la dernière implication on utilise que \mathbb{Z}/n est un corps. \square

L'astuce de la racine carrée

Si n est premier, alors tout $x \in \mathbb{Z}/n^*$ vérifie $x^{n-1} = 1$.

Or, ce beau critère n'est pas infallible. Essayons de l'optimiser :

Critère

Si n est un nombre premier impair, alors tout $x \in \mathbb{Z}/n^$ vérifie $x^{\frac{n-1}{2}} = \pm 1$. \square*

Démonstration. D'après Fermat $y = x^{\frac{n-1}{2}}$ vérifie $y^2 = 1$.

Si n est premier, alors $y^2 = 1$ n'admet que deux solutions dans \mathbb{Z}/n :

$$y^2 = 1$$

$$\Leftrightarrow y^2 - 1 = 0$$

$$\Leftrightarrow (y - 1)(y + 1) = 0$$

$$\Leftrightarrow y = 1 \text{ ou } y = -1.$$

Pour la dernière implication on utilise que \mathbb{Z}/n est un corps. \square

Remarque

Si $n = pq$ est composé de deux facteurs premiers distincts, alors $y^2 = 1$ admet quatre solutions : dans $\mathbb{Z}/n \cong \mathbb{Z}/p \times \mathbb{Z}/q$ on trouve les deux $(\bar{1}, \bar{1})$ et $(-\bar{1}, -\bar{1})$ comme avant, mais aussi deux autres $(-\bar{1}, +\bar{1})$ et $(+\bar{1}, -\bar{1})$.

Critère (de Miller–Rabin)

Soit $n \in \mathbb{N}$ impair. On décompose $n - 1 = 2^e q$ avec $e \geq 1$ et q impair.

Critère (de Miller–Rabin)

Soit $n \in \mathbb{N}$ impair. On décompose $n - 1 = 2^e q$ avec $e \geq 1$ et q impair.

Si n est premier, alors tout $x \in \mathbb{Z}/n^$ satisfait ou $x^q = 1$*

ou il existe $k \in \{0, \dots, e - 1\}$ tel que $x^{2^k q} = -1$.

Critère (de Miller–Rabin)

Soit $n \in \mathbb{N}$ impair. On décompose $n - 1 = 2^e q$ avec $e \geq 1$ et q impair.

Si n est premier, alors tout $x \in \mathbb{Z}/n^$ satisfait ou $x^q = 1$*

ou il existe $k \in \{0, \dots, e - 1\}$ tel que $x^{2^k q} = -1$.

Démonstration. D'après Fermat $y = x^q$ vérifie $y^{2^e} = 1$.

Critère (de Miller–Rabin)

Soit $n \in \mathbb{N}$ impair. On décompose $n - 1 = 2^e q$ avec $e \geq 1$ et q impair.

Si n est premier, alors tout $x \in \mathbb{Z}/n^$ satisfait ou $x^q = 1$*

ou il existe $k \in \{0, \dots, e - 1\}$ tel que $x^{2^k q} = -1$.

Démonstration. D'après Fermat $y = x^q$ vérifie $y^{2^e} = 1$.

Pour $k := \min\{i \in \mathbb{N} \mid y^{2^i} = 1\}$ on a donc $k \leq e$.

Critère (de Miller–Rabin)

Soit $n \in \mathbb{N}$ impair. On décompose $n - 1 = 2^e q$ avec $e \geq 1$ et q impair.

Si n est premier, alors tout $x \in \mathbb{Z}/n^$ satisfait ou $x^q = 1$*

ou il existe $k \in \{0, \dots, e - 1\}$ tel que $x^{2^k q} = -1$.

Démonstration. D'après Fermat $y = x^q$ vérifie $y^{2^e} = 1$.

Pour $k := \min\{i \in \mathbb{N} \mid y^{2^i} = 1\}$ on a donc $k \leq e$.

Si $k > 1$ alors $y^{2^k} = 1$ et $y^{2^{k-1}} = -1$.

Critère (de Miller–Rabin)

Soit $n \in \mathbb{N}$ impair. On décompose $n - 1 = 2^e q$ avec $e \geq 1$ et q impair.

Si n est premier, alors tout $x \in \mathbb{Z}/n^$ satisfait ou $x^q = 1$*

ou il existe $k \in \{0, \dots, e - 1\}$ tel que $x^{2^k q} = -1$.

Démonstration. D'après Fermat $y = x^q$ vérifie $y^{2^e} = 1$.

Pour $k := \min\{i \in \mathbb{N} \mid y^{2^i} = 1\}$ on a donc $k \leq e$.

Si $k > 1$ alors $y^{2^k} = 1$ et $y^{2^{k-1}} = -1$.

Si $k = 0$ alors $y = 1$.



Critère (de Miller–Rabin)

Soit $n \in \mathbb{N}$ impair. On décompose $n - 1 = 2^e q$ avec $e \geq 1$ et q impair.

Si n est premier, alors tout $x \in \mathbb{Z}/n^*$ satisfait ou $x^q = 1$

ou il existe $k \in \{0, \dots, e - 1\}$ tel que $x^{2^k q} = -1$.

Démonstration. D'après Fermat $y = x^q$ vérifie $y^{2^e} = 1$.

Pour $k := \min\{i \in \mathbb{N} \mid y^{2^i} = 1\}$ on a donc $k \leq e$.

Si $k > 1$ alors $y^{2^k} = 1$ et $y^{2^{k-1}} = -1$.

Si $k = 0$ alors $y = 1$.



Test de primalité selon Miller–Rabin :

Critère (de Miller–Rabin)

Soit $n \in \mathbb{N}$ impair. On décompose $n - 1 = 2^e q$ avec $e \geq 1$ et q impair.

Si n est premier, alors tout $x \in \mathbb{Z}/n^*$ satisfait ou $x^q = 1$

ou il existe $k \in \{0, \dots, e - 1\}$ tel que $x^{2^k q} = -1$.

Démonstration. D'après Fermat $y = x^q$ vérifie $y^{2^e} = 1$.

Pour $k := \min\{i \in \mathbb{N} \mid y^{2^i} = 1\}$ on a donc $k \leq e$.

Si $k > 1$ alors $y^{2^k} = 1$ et $y^{2^{k-1}} = -1$.

Si $k = 0$ alors $y = 1$.



Test de primalité selon Miller–Rabin :

On tire x au hasard dans \mathbb{Z}/n^* puis on vérifie le critère ci-dessus.

Critère (de Miller–Rabin)

Soit $n \in \mathbb{N}$ impair. On décompose $n - 1 = 2^e q$ avec $e \geq 1$ et q impair.

Si n est premier, alors tout $x \in \mathbb{Z}/n^*$ satisfait ou $x^q = 1$

ou il existe $k \in \{0, \dots, e - 1\}$ tel que $x^{2^k q} = -1$.

Démonstration. D'après Fermat $y = x^q$ vérifie $y^{2^e} = 1$.

Pour $k := \min\{i \in \mathbb{N} \mid y^{2^i} = 1\}$ on a donc $k \leq e$.

Si $k > 1$ alors $y^{2^k} = 1$ et $y^{2^{k-1}} = -1$.

Si $k = 0$ alors $y = 1$.



Test de primalité selon Miller–Rabin :

On tire x au hasard dans \mathbb{Z}/n^* puis on vérifie le critère ci-dessus.

Si $x^q \neq \bar{1}$ et $x^{2^k q} \neq -\bar{1}$ pour $k = 0, \dots, e - 1$, alors n est composé.

Critère (de Miller–Rabin)

Soit $n \in \mathbb{N}$ impair. On décompose $n - 1 = 2^e q$ avec $e \geq 1$ et q impair.

Si n est premier, alors tout $x \in \mathbb{Z}/n^*$ satisfait ou $x^q = 1$

ou il existe $k \in \{0, \dots, e - 1\}$ tel que $x^{2^k q} = -1$.

Démonstration. D'après Fermat $y = x^q$ vérifie $y^{2^e} = 1$.

Pour $k := \min\{i \in \mathbb{N} \mid y^{2^i} = 1\}$ on a donc $k \leq e$.

Si $k > 1$ alors $y^{2^k} = 1$ et $y^{2^{k-1}} = -1$.

Si $k = 0$ alors $y = 1$.



Test de primalité selon Miller–Rabin :

On tire x au hasard dans \mathbb{Z}/n^* puis on vérifie le critère ci-dessus.

Si $x^q \neq \bar{1}$ et $x^{2^k q} \neq -\bar{1}$ pour $k = 0, \dots, e - 1$, alors n est composé.

Dans ce cas on dit que x est un *témoin de décomposabilité* de n ,
ou que x *témoigne* contre la primalité de n (au sens de Miller–Rabin).

Critère (de Miller–Rabin)

Soit $n \in \mathbb{N}$ impair. On décompose $n - 1 = 2^e q$ avec $e \geq 1$ et q impair.

Si n est premier, alors tout $x \in \mathbb{Z}/n^*$ satisfait ou $x^q = 1$

ou il existe $k \in \{0, \dots, e - 1\}$ tel que $x^{2^k q} = -1$.

Démonstration. D'après Fermat $y = x^q$ vérifie $y^{2^e} = 1$.

Pour $k := \min\{i \in \mathbb{N} \mid y^{2^i} = 1\}$ on a donc $k \leq e$.

Si $k > 1$ alors $y^{2^k} = 1$ et $y^{2^{k-1}} = -1$.

Si $k = 0$ alors $y = 1$. □

Test de primalité selon Miller–Rabin :

On tire x au hasard dans \mathbb{Z}/n^* puis on vérifie le critère ci-dessus.

Si $x^q \neq \bar{1}$ et $x^{2^k q} \neq -\bar{1}$ pour $k = 0, \dots, e - 1$, alors n est composé.

Dans ce cas on dit que x est un *témoin de décomposabilité* de n ,
ou que x *témoigne* contre la primalité de n (au sens de Miller–Rabin).

Le critère est nécessaire pour la primalité de n mais non suffisante.

Critère (de Miller–Rabin)

Soit $n \in \mathbb{N}$ impair. On décompose $n - 1 = 2^e q$ avec $e \geq 1$ et q impair.

Si n est premier, alors tout $x \in \mathbb{Z}/n^*$ satisfait ou $x^q = 1$

ou il existe $k \in \{0, \dots, e - 1\}$ tel que $x^{2^k q} = -1$.

Démonstration. D'après Fermat $y = x^q$ vérifie $y^{2^e} = 1$.

Pour $k := \min\{i \in \mathbb{N} \mid y^{2^i} = 1\}$ on a donc $k \leq e$.

Si $k > 1$ alors $y^{2^k} = 1$ et $y^{2^{k-1}} = -1$.

Si $k = 0$ alors $y = 1$. □

Test de primalité selon Miller–Rabin :

On tire x au hasard dans \mathbb{Z}/n^* puis on vérifie le critère ci-dessus.

Si $x^q \neq \bar{1}$ et $x^{2^k q} \neq -\bar{1}$ pour $k = 0, \dots, e - 1$, alors n est composé.

Dans ce cas on dit que x est un *témoin de décomposabilité* de n ,
ou que x *témoigne* contre la primalité de n (au sens de Miller–Rabin).

Le critère est nécessaire pour la primalité de n mais non suffisante.

Si x passe le critère on dit que n est *pseudo-premier* à base x .

Algorithme 6.5 le critère de Miller–Rabin

Entrée: un entier impair $n \geq 5$ et un entier x

Sortie: le message « composé » ou « pseudo-premier »

```
décomposer  $n - 1 = 2^e q$  où  $e \geq 1$  et  $q$  impair // divisions itérées
calculer  $y \leftarrow x^q \bmod n$  // puissance modulaire rapide
si  $y = 1$  alors retourner « pseudo-premier » //  $x$  passe car  $x^q \equiv 1$ .
pour  $k$  de 1 à  $e$  faire
    si  $y = n - 1$  alors retourner « pseudo-premier » //  $x$  passe car  $x^{2^{k-1}q} \equiv -1$ 
     $y \leftarrow y^2 \bmod n$  // calculer  $y = x^{2^k q} \bmod n$ 
fin pour
retourner « composé » //  $x$  ne passe pas le critère.
```

Algorithme 6.6 le critère de Miller–Rabin

Entrée: un entier impair $n \geq 5$ et un entier x

Sortie: le message « composé » ou « pseudo-premier »

```
décomposer  $n - 1 = 2^e q$  où  $e \geq 1$  et  $q$  impair // divisions itérées
calculer  $y \leftarrow x^q \bmod n$  // puissance modulaire rapide
si  $y = 1$  alors retourner « pseudo-premier » //  $x$  passe car  $x^q \equiv 1$ .
pour  $k$  de 1 à  $e$  faire
    si  $y = n - 1$  alors retourner « pseudo-premier » //  $x$  passe car  $x^{2^{k-1}q} \equiv -1$ 
     $y \leftarrow y^2 \bmod n$  // calculer  $y = x^{2^k q} \bmod n$ 
fin pour
retourner « composé » //  $x$  ne passe pas le critère.
```

Exercice

Quelle est la complexité de cet algorithme ?
Comment interpréter les deux réponses ?

Exemple

Explicitons les témoins $x \in \mathbb{Z}/n^\times$ pour $n = 21$.

Exemple

Explicitons les témoins $x \in \mathbb{Z}/n^\times$ pour $n = 21$.

Ici $n - 1 = 20 = 2^2 \cdot 5$ donc $e = 2$ et $q = 5$.

Témoins au sens de Fermat et de Miller–Rabin

Exemple

Explicitons les témoins $x \in \mathbb{Z}/n^{\times}$ pour $n = 21$.

Ici $n - 1 = 20 = 2^2 \cdot 5$ donc $e = 2$ et $q = 5$.

x	x^5	x^{10}	x^{20}	critère	x	x^5	x^{10}	x^{20}	critère
1	1	1	1	jamais	11	2	4	16	Fermat
2	11	16	4	Fermat	12	3	9	18	pgcd
3	12	18	9	pgcd	13	13	1	1	Miller–Rabin
4	16	4	16	Fermat	14	14	7	7	pgcd
5	17	16	4	Fermat	15	15	15	15	pgcd
6	6	15	15	pgcd	16	4	16	4	Fermat
7	7	7	7	pgcd	17	5	4	16	Fermat
8	8	1	1	Miller–Rabin	18	9	18	9	pgcd
9	18	9	18	pgcd	19	10	16	4	Fermat
10	19	4	16	Fermat	20	20	1	1	jamais

Théorème (Rabin et Monier 1980)

Soit n un entier impair, $n \geq 3$.

- 1** *Si n est premier, alors tout $x \in \mathbb{Z}/n^*$ satisfait au critère de Miller–Rabin.*
- 2** *Si n est composé, alors il y a au moins $\frac{3}{4}(n - 1)$ témoins $x \in \mathbb{Z}/n^*$.*

Théorème (Rabin et Monier 1980)

Soit n un entier impair, $n \geq 3$.

- 1 Si n est premier, alors tout $x \in \mathbb{Z}/n^*$ satisfait au critère de Miller–Rabin.
- 2 Si n est composé, alors il y a au moins $\frac{3}{4}(n - 1)$ témoins $x \in \mathbb{Z}/n^*$.

Il y a au moins 75% de témoins et au plus 25% de non-témoins.

Théorème (Rabin et Monier 1980)

Soit n un entier impair, $n \geq 3$.

- 1 Si n est premier, alors tout $x \in \mathbb{Z}/n^*$ satisfait au critère de Miller–Rabin.
- 2 Si n est composé, alors il y a au moins $\frac{3}{4}(n - 1)$ témoins $x \in \mathbb{Z}/n^*$.

Il y a au moins 75% de témoins et au plus 25% de non-témoins.

En général les témoins sont encore beaucoup plus fréquents.

Théorème (Rabin et Monier 1980)

Soit n un entier impair, $n \geq 3$.

- 1 Si n est premier, alors tout $x \in \mathbb{Z}/n^*$ satisfait au critère de Miller–Rabin.
- 2 Si n est composé, alors il y a au moins $\frac{3}{4}(n - 1)$ témoins $x \in \mathbb{Z}/n^*$.

Il y a au moins 75% de témoins et au plus 25% de non-témoins.

En général les témoins sont encore beaucoup plus fréquents.

Exemple

Reprenons le plus petit nombre de Carmichael, $n = 561 = 3 \cdot 11 \cdot 17$.

Théorème (Rabin et Monier 1980)

Soit n un entier impair, $n \geq 3$.

- 1 Si n est premier, alors tout $x \in \mathbb{Z}/n^*$ satisfait au critère de Miller–Rabin.
- 2 Si n est composé, alors il y a au moins $\frac{3}{4}(n - 1)$ témoins $x \in \mathbb{Z}/n^*$.

Il y a au moins 75% de témoins et au plus 25% de non-témoins.

En général les témoins sont encore beaucoup plus fréquents.

Exemple

Reprenons le plus petit nombre de Carmichael, $n = 561 = 3 \cdot 11 \cdot 17$.

Parmi les 560 éléments non nuls, 240 ne sont pas inversibles (43%).

Théorème (Rabin et Monier 1980)

Soit n un entier impair, $n \geq 3$.

- 1 Si n est premier, alors tout $x \in \mathbb{Z}/n^*$ satisfait au critère de Miller–Rabin.
- 2 Si n est composé, alors il y a au moins $\frac{3}{4}(n - 1)$ témoins $x \in \mathbb{Z}/n^*$.

Il y a au moins 75% de témoins et au plus 25% de non-témoins.

En général les témoins sont encore beaucoup plus fréquents.

Exemple

Reprenons le plus petit nombre de Carmichael, $n = 561 = 3 \cdot 11 \cdot 17$.

Parmi les 560 éléments non nuls, 240 ne sont pas inversibles (43%).

Tous les éléments inversibles satisfont au critère de Fermat.

Théorème (Rabin et Monier 1980)

Soit n un entier impair, $n \geq 3$.

- 1 Si n est premier, alors tout $x \in \mathbb{Z}/n^*$ satisfait au critère de Miller–Rabin.
- 2 Si n est composé, alors il y a au moins $\frac{3}{4}(n - 1)$ témoins $x \in \mathbb{Z}/n^*$.

Il y a au moins 75% de témoins et au plus 25% de non-témoins.

En général les témoins sont encore beaucoup plus fréquents.

Exemple

Reprenons le plus petit nombre de Carmichael, $n = 561 = 3 \cdot 11 \cdot 17$.

Parmi les 560 éléments non nuls, 240 ne sont pas inversibles (43%).

Tous les éléments inversibles satisfont au critère de Fermat.

Seulement 10 éléments passent le critère de Miller–Rabin, soit moins de 2%.
(Il s'agit de 1, 50, 101, 103, 256, 305, 458, 460, 511, 560.)

Algorithme 6.7 test de primalité selon Miller–Rabin

Entrée: un entier impair $n \geq 5$

Sortie: le message « composé » ou « pseudo-premier »

décomposer $n - 1 = 2^e q$ où $e \geq 1$ et q impair // divisions itérées

pour t **de** 1 **à** 50 **faire**

 tirer $x \in \mathbb{Z}/_n^*$ au hasard et appliquer le critère de Miller–Rabin

si x témoigne que n est composé **alors retourner** « composé »

fin pour

retourner « pseudo-premier »

Algorithme 6.8 test de primalité selon Miller–Rabin

Entrée: un entier impair $n \geq 5$

Sortie: le message « composé » ou « pseudo-premier »

décomposer $n - 1 = 2^e q$ où $e \geq 1$ et q impair // divisions itérées

pour t **de** 1 **à** 50 **faire**

 tirer $x \in \mathbb{Z}/n^*$ au hasard et appliquer le critère de Miller–Rabin

si x témoigne que n est composé **alors retourner** « composé »

fin pour

retourner « pseudo-premier »

Proposition

Soit $n \in \mathbb{N}$ un entier impair, $n \geq 5$.

Algorithme 6.9 test de primalité selon Miller–Rabin

Entrée: un entier impair $n \geq 5$

Sortie: le message « composé » ou « pseudo-premier »

décomposer $n - 1 = 2^e q$ où $e \geq 1$ et q impair // divisions itérées

pour t **de** 1 **à** 50 **faire**

 tirer $x \in \mathbb{Z}/n^*$ au hasard et appliquer le critère de Miller–Rabin

si x témoigne que n est composé **alors retourner** « composé »

fin pour

retourner « pseudo-premier »

Proposition

Soit $n \in \mathbb{N}$ un entier impair, $n \geq 5$.

- 1 Si n est premier, alors l'algorithme répond toujours « pseudo-premier ».

Algorithme 6.10 test de primalité selon Miller–Rabin

Entrée: un entier impair $n \geq 5$

Sortie: le message « composé » ou « pseudo-premier »

décomposer $n - 1 = 2^e q$ où $e \geq 1$ et q impair // divisions itérées

pour t **de** 1 **à** 50 **faire**

 tirer $x \in \mathbb{Z}/n^*$ au hasard et appliquer le critère de Miller–Rabin

si x témoigne que n est composé **alors retourner** « composé »

fin pour

retourner « pseudo-premier »

Proposition

Soit $n \in \mathbb{N}$ un entier impair, $n \geq 5$.

- 1 Si n est premier, alors l'algorithme répond toujours « pseudo-premier ».
- 2 Si n est composé, alors les deux réponses sont possibles :

Algorithme 6.11 test de primalité selon Miller–Rabin

Entrée: un entier impair $n \geq 5$

Sortie: le message « composé » ou « pseudo-premier »

décomposer $n - 1 = 2^e q$ où $e \geq 1$ et q impair // divisions itérées

pour t **de** 1 **à** 50 **faire**

 tirer $x \in \mathbb{Z}/n^*$ au hasard et appliquer le critère de Miller–Rabin

si x témoigne que n est composé **alors retourner** « composé »

fin pour

retourner « pseudo-premier »

Proposition

Soit $n \in \mathbb{N}$ un entier impair, $n \geq 5$.

- 1 Si n est premier, alors l'algorithme répond toujours « pseudo-premier ».
- 2 Si n est composé, alors les deux réponses sont possibles :
L'algorithme répond « pseudo-premier » avec une probabilité $\leq 4^{-50}$.

Algorithme 6.12 test de primalité selon Miller–Rabin

Entrée: un entier impair $n \geq 5$

Sortie: le message « composé » ou « pseudo-premier »

décomposer $n - 1 = 2^e q$ où $e \geq 1$ et q impair // divisions itérées

pour t **de** 1 **à** 50 **faire**

 tirer $x \in \mathbb{Z}/n^*$ au hasard et appliquer le critère de Miller–Rabin

si x témoigne que n est composé **alors retourner** « composé »

fin pour

retourner « pseudo-premier »

Proposition

Soit $n \in \mathbb{N}$ un entier impair, $n \geq 5$.

- 1 Si n est premier, alors l'algorithme répond toujours « pseudo-premier ».
- 2 Si n est composé, alors les deux réponses sont possibles :
L'algorithme répond « pseudo-premier » avec une probabilité $\leq 4^{-50}$.
L'algorithme répond « composé » avec une probabilité $\geq 1 - 4^{-50}$.

Algorithme 6.13 production de grands nombres premiers aléatoires

Entrée: un entier impair $n \in \mathbb{N}$ vérifiant $n \geq 3$.

Sortie: un nombre premier aléatoire $p \in \mathbb{N}$ vérifiant $n \leq p < 2n$

$p \leftarrow 2k + 1$ où l'on choisit k aléatoirement dans $\{\frac{n-1}{2}, \dots, n-1\}$

si p admet un petit facteur 3, 5, 7, 11, ... **alors** recommencer

si le test de Miller–Rabin sur p répond « composé » **alors** recommencer

retourner p

Algorithme 6.14 production de grands nombres premiers aléatoires

Entrée: un entier impair $n \in \mathbb{N}$ vérifiant $n \geq 3$.

Sortie: un nombre premier aléatoire $p \in \mathbb{N}$ vérifiant $n \leq p < 2n$

$p \leftarrow 2k + 1$ où l'on choisit k aléatoirement dans $\{\frac{n-1}{2}, \dots, n-1\}$

si p admet un petit facteur 3, 5, 7, 11, ... **alors** recommencer

si le test de Miller–Rabin sur p répond « composé » **alors** recommencer

retourner p

Terminaison : La probabilité que p soit premier est $\sim 2/\ln(n)$.

Algorithme 6.15 production de grands nombres premiers aléatoires

Entrée: un entier impair $n \in \mathbb{N}$ vérifiant $n \geq 3$.

Sortie: un nombre premier aléatoire $p \in \mathbb{N}$ vérifiant $n \leq p < 2n$

$p \leftarrow 2k + 1$ où l'on choisit k aléatoirement dans $\{\frac{n-1}{2}, \dots, n-1\}$

si p admet un petit facteur 3, 5, 7, 11, ... **alors** recommencer

si le test de Miller–Rabin sur p répond « composé » **alors** recommencer

retourner p

Terminaison : La probabilité que p soit premier est $\sim 2/\ln(n)$.

En moyenne il faut itérer $\sim \ln(n)/2$ fois avant de tomber sur un premier.

Algorithme 6.16 production de grands nombres premiers aléatoires

Entrée: un entier impair $n \in \mathbb{N}$ vérifiant $n \geq 3$.

Sortie: un nombre premier aléatoire $p \in \mathbb{N}$ vérifiant $n \leq p < 2n$

$p \leftarrow 2k + 1$ où l'on choisit k aléatoirement dans $\{\frac{n-1}{2}, \dots, n-1\}$

si p admet un petit facteur 3, 5, 7, 11, ... **alors** recommencer

si le test de Miller–Rabin sur p répond « composé » **alors** recommencer

retourner p

Terminaison : La probabilité que p soit premier est $\sim 2/\ln(n)$.

En moyenne il faut itérer $\sim \ln(n)/2$ fois avant de tomber sur un premier.

Probabilité d'erreur : Supposons que $\ln(n) \ll 4^{50}$, disons $\ln(n) \leq 4^{10}$.

Algorithme 6.17 production de grands nombres premiers aléatoires

Entrée: un entier impair $n \in \mathbb{N}$ vérifiant $n \geq 3$.

Sortie: un nombre premier aléatoire $p \in \mathbb{N}$ vérifiant $n \leq p < 2n$

$p \leftarrow 2k + 1$ où l'on choisit k aléatoirement dans $\{\frac{n-1}{2}, \dots, n-1\}$

si p admet un petit facteur 3, 5, 7, 11, ... **alors** recommencer

si le test de Miller–Rabin sur p répond « composé » **alors** recommencer

retourner p

Terminaison : La probabilité que p soit premier est $\sim 2/\ln(n)$.

En moyenne il faut itérer $\sim \ln(n)/2$ fois avant de tomber sur un premier.

Probabilité d'erreur : Supposons que $\ln(n) \ll 4^{50}$, disons $\ln(n) \leq 4^{10}$.

Alors la probabilité que p soit composé est $\leq 4^{-50} \ln(n)$.

Algorithme 6.18 production de grands nombres premiers aléatoires

Entrée: un entier impair $n \in \mathbb{N}$ vérifiant $n \geq 3$.

Sortie: un nombre premier aléatoire $p \in \mathbb{N}$ vérifiant $n \leq p < 2n$

$p \leftarrow 2k + 1$ où l'on choisit k aléatoirement dans $\{\frac{n-1}{2}, \dots, n-1\}$

si p admet un petit facteur 3, 5, 7, 11, ... **alors** recommencer

si le test de Miller–Rabin sur p répond « composé » **alors** recommencer

retourner p

Terminaison : La probabilité que p soit premier est $\sim 2/\ln(n)$.

En moyenne il faut itérer $\sim \ln(n)/2$ fois avant de tomber sur un premier.

Probabilité d'erreur : Supposons que $\ln(n) \ll 4^{50}$, disons $\ln(n) \leq 4^{10}$.

Alors la probabilité que p soit composé est $\leq 4^{-50} \ln(n)$.

Précautions pratiques :

Algorithme 6.19 production de grands nombres premiers aléatoires

Entrée: un entier impair $n \in \mathbb{N}$ vérifiant $n \geq 3$.

Sortie: un nombre premier aléatoire $p \in \mathbb{N}$ vérifiant $n \leq p < 2n$

$p \leftarrow 2k + 1$ où l'on choisit k aléatoirement dans $\{\frac{n-1}{2}, \dots, n-1\}$

si p admet un petit facteur 3, 5, 7, 11, ... **alors** recommencer

si le test de Miller–Rabin sur p répond « composé » **alors** recommencer

retourner p

Terminaison : La probabilité que p soit premier est $\sim 2/\ln(n)$.

En moyenne il faut itérer $\sim \ln(n)/2$ fois avant de tomber sur un premier.

Probabilité d'erreur : Supposons que $\ln(n) \ll 4^{50}$, disons $\ln(n) \leq 4^{10}$.

Alors la probabilité que p soit composé est $\leq 4^{-50} \ln(n)$.

Précautions pratiques :

- Il faut un bon générateur de nombres aléatoires (voir Knuth).

Algorithme 6.20 production de grands nombres premiers aléatoires

Entrée: un entier impair $n \in \mathbb{N}$ vérifiant $n \geq 3$.

Sortie: un nombre premier aléatoire $p \in \mathbb{N}$ vérifiant $n \leq p < 2n$

$p \leftarrow 2k + 1$ où l'on choisit k aléatoirement dans $\{\frac{n-1}{2}, \dots, n-1\}$

si p admet un petit facteur 3, 5, 7, 11, ... **alors** recommencer

si le test de Miller–Rabin sur p répond « composé » **alors** recommencer

retourner p

Terminaison : La probabilité que p soit premier est $\sim 2/\ln(n)$.

En moyenne il faut itérer $\sim \ln(n)/2$ fois avant de tomber sur un premier.

Probabilité d'erreur : Supposons que $\ln(n) \ll 4^{50}$, disons $\ln(n) \leq 4^{10}$.

Alors la probabilité que p soit composé est $\leq 4^{-50} \ln(n)$.

Précautions pratiques :

- Il faut un bon générateur de nombres aléatoires (voir Knuth).
- Pour RSA il faut encore éviter des nombres premiers « faibles ».