Lecture in Number Theory

College of Science for Women

Baghdad University

March 31, 2014

Factoring integers, Producing primes and the RSA cryptosystem

Francesco Pappalardi

How large are large numbers?

How large are large numbers?

☞

☞

☞

☞

☞

How large are large numbers?

- NUMBER OF CELLS IN A HUMAN BODY: 10^{15}
-
-
-
-

How large are large numbers?

- NUMBER OF CELLS IN A HUMAN BODY: 10^{15}
- NUMBER OF ATOMS IN THE UNIVERSE: 10^{80}
-
-
-

How large are large numbers?

- ☞ NUMBER OF CELLS IN A HUMAN BODY: 10^{15}
- ☞ NUMBER OF ATOMS IN THE UNIVERSE: 10^{80}
- ☞ NUMBER OF SUBATOMIC PARTICLES IN THE UNIVERSE: 10^{120}
- ☞
- ☞

How large are large numbers?

- number of cells in a human body: 10^{15}
- number of atoms in the universe: 10^{80}
- number of subatomic particles in the universe: 10^{120}
- number of atoms in a Human Brain: 10^{27}
-

How large are large numbers?

- ☞ NUMBER OF CELLS IN A HUMAN BODY: 10^{15}
- ☞ NUMBER OF ATOMS IN THE UNIVERSE: 10^{80}
- ☞ NUMBER OF SUBATOMIC PARTICLES IN THE UNIVERSE: 10^{120}
- ☞ NUMBER OF ATOMS IN A HUMAN BRAIN: 10^{27}
- ☞ NUMBER OF ATOMS IN A CAT: 10^{26}

$$
\begin{aligned}
RSA_{2048} = {} & 25195908475657893494027183240048398571429282126204 \\
& 03202777713783604366202070759555626401852588078440691829064124 9 \\
& 5150821892985591491761845028084891200728449926873928072877767 35 \\
& 9714183472702618963750149718246911650776133798590957000973304 59 \\
& 7488084284017974291006424586918171951187461215151726546322822 16 \\
& 8699875491824224336372590851418654620435767984233871847744479 20 \\
& 7399342365848238242811981638150106748104516603773060562016196 76 \\
& 2561338441436038339044149526344321901146575444541784240209246 16 \\
& 5157233507787077498171257724679629263863563732899121548314381 67 \\
& 8998850404453640235273819513786365643912120103971228221207203 57
\end{aligned}
$$

RSA_{2048} = 25195908475657893494027183240048398571429282126204
03202777713783604366202070759555626401852588078440691829064124
95150821892985591491761845028084891200728449926873928072877767
35971418347270261896375014971824691165077613379859095700097330
459748808428401797429100642458691817195118746121515172654632282216
86998754918242243363725908514186546204357679842338718477444792
07399342365848238242811981638150106748104516603773060562016196
76256133844143603833904414952634432190114657544454178424020924616
51572335077870774981712577246796292638635637328991215483143816
7899885040445364023527381951378636564391212010397122822120720357

RSA_{2048} is a 617 (decimal) digit number

RSA_{2048} = 25195908475657893494027183240048398571429282126204
032027777137836043662020707595556264018525880784406918290641249
515082189298559149176184502808489120072844992687392807287776735
971418347270261896375014971824691165077613379859095700097330459
748808428401797429100642458691817195118746121515172654632282216
869987549182422433637259085141865462043576798423387184774447920
739934236584823824281198163815010674810451660377306056201619676
256133844143603833904414952634432190114657544454178424020924616
515723350778707749817125772467962926386356373289912154831438167
899885040445364023527381951378636564391212010397122822120720357

RSA_{2048} is a 617 (decimal) digit number

http://www.rsa.com/rsalabs/challenges/factoring/numbers.html/

$$RSA_{2048} = p \cdot q, \quad p, q \approx 10^{308}$$

$$RSA_{2048} = p \cdot q, \quad p, q \approx 10^{308}$$

PROBLEM: *Compute p and q*

$$RSA_{2048} = p \cdot q, \quad p, q \approx 10^{308}$$

PROBLEM: *Compute p and q*

PRICE OFFERED ON MARCH 18, 1991: 200.000 US$ ($\sim$ 232.700.000 Iraq Dinars)!!

$$RSA_{2048} = p \cdot q, \quad p, q \approx 10^{308}$$

PROBLEM: *Compute p and q*

PRICE OFFERED ON MARCH 18, 1991: 200.000 US$ ($\sim$ 232.700.000 Iraq Dinars)!!

Theorem. If $a \in \mathbb{N} \quad \exists!\ p_1 < p_2 < \cdots < p_k$ *primes*

$$\text{s.t.} \quad a = p_1^{\alpha_1} \cdots p_k^{\alpha_k}$$

$$RSA_{2048}{=}p\cdot q,\qquad p,q\approx 10^{308}$$

PROBLEM: *Compute p and q*

PRICE OFFERED ON MARCH 18, 1991: 200.000 US$ (∼ 232.700.000 Iraq Dinars)!!

Theorem. If $a\in\mathbb{N}\quad\exists!\ p_1<p_2<\cdots<p_k$ *primes*

s.t. $a=p_1^{\alpha_1}\cdots p_k^{\alpha_k}$

Regrettably: RSAlabs believes that factoring in one year requires:

number	computers	memory
RSA_{1620}	1.6×10^{15}	120 Tb
RSA_{1024}	$342,000,000$	170 Gb
RSA_{760}	215,000	4Gb.

http://www.rsa.com/rsalabs/challenges/factoring/numbers.html

http://www.rsa.com/rsalabs/challenges/factoring/numbers.html

Challenge Number	Prize ($US)
RSA_{576}	$10,000
RSA_{640}	$20,000
RSA_{704}	$30,000
RSA_{768}	$50,000
RSA_{896}	$75,000
RSA_{1024}	$100,000
RSA_{1536}	$150,000
RSA_{2048}	$200,000

http://www.rsa.com/rsalabs/challenges/factoring/numbers.html

Numero	Premio ($US)	Status
RSA_{576}	$10,000	Factored December 2003
RSA_{640}	$20,000	Factored November 2005
RSA_{704}	$30,000	Factored July, 2 2012
RSA_{768}	$50,000	Factored December, 12 2009
RSA_{896}	$75,000	Not factored
RSA_{1024}	$100,000	Not factored
RSA_{1536}	$150,000	Not factored
RSA_{2048}	$200,000	Not factored

http://www.rsa.com/rsalabs/challenges/factoring/numbers.html

Numero	Premio ($US)	Status
RSA_{576}	$10,000	Factored December 2003
RSA_{640}	$20,000	Factored November 2005
RSA_{704}	$30,000	Factored July, 2 2012
RSA_{768}	$50,000	Factored December, 12 2009
RSA_{896}	$75,000	Not factored
RSA_{1024}	$100,000	Not factored
RSA_{1536}	$150,000	Not factored
RSA_{2048}	$200,000	Not factored

The RSA challenges ended in 2007. RSA Laboratories stated:
"*Now that the industry has a considerably more advanced understanding of the cryptanalytic strength of common symmetric-key and public-key algorithms, these challenges are no longer active.*"

Famous citation!!!

A phenomenon whose probability is 10^{-50} *never happens, and it will never observed.*

- Émil Borel (Les probabilités et sa vie)

History of the "Art of Factoring"

History of the "Art of Factoring"

- 220 BC Greeks (Eratosthenes of Cyrene)

History of the "Art of Factoring"

- 220 BC Greeks (Eratosthenes of Cyrene)
- 1730 Euler $2^{2^5}+1=641\cdot 6700417$

History of the "Art of Factoring"

- 220 BC Greeks (Eratosthenes of Cyrene)
- 1730 Euler $2^{2^5} + 1 = 641 \cdot 6700417$
- 1750–1800 Fermat, Gauss (Sieves - Tables)

History of the "Art of Factoring"

- 220 BC Greeks (Eratosthenes of Cyrene)
- 1730 Euler $2^{2^5} + 1 = 641 \cdot 6700417$
- 1750–1800 Fermat, Gauss (Sieves - Tables)
- 1880 Landry & Le Lasseur:

$$2^{2^6} + 1 = 274177 \times 67280421310721$$

History of the "Art of Factoring"

- 220 BC Greeks (Eratosthenes of Cyrene)
- 1730 Euler $2^{2^5} + 1 = 641 \cdot 6700417$
- 1750–1800 Fermat, Gauss (Sieves - Tables)
- 1880 Landry & Le Lasseur:

$$2^{2^6} + 1 = 274177 \times 67280421310721$$

- 1919 Pierre and Eugène Carissan (Factoring Machine)

History of the "Art of Factoring"

- 220 BC Greeks (Eratosthenes of Cyrene)
- 1730 Euler $2^{2^5} + 1 = 641 \cdot 6700417$
- 1750–1800 Fermat, Gauss (Sieves - Tables)
- 1880 Landry & Le Lasseur:

$$2^{2^6} + 1 = 274177 \times 67280421310721$$

- 1919 Pierre and Eugène Carissan (Factoring Machine)
- 1970 Morrison & Brillhart

$$2^{2^7} + 1 = 59649589127497217 \times 5704689200685129054721$$

History of the "Art of Factoring"

- 220 BC Greeks (Eratosthenes of Cyrene)
- 1730 Euler $2^{2^5}+1=641\cdot 6700417$
- 1750–1800 Fermat, Gauss (Sieves - Tables)
- 1880 Landry & Le Lasseur:

$$2^{2^6}+1=274177\times 67280421310721$$

- 1919 Pierre and Eugène Carissan (Factoring Machine)
- 1970 Morrison & Brillhart

$$2^{2^7}+1=59649589127497217\times 5704689200685129054721$$

- 1982 Quadratic Sieve **QS** (Pomerance) $\rightsquigarrow$ Number Fields Sieve **NFS**

History of the "Art of Factoring"

- 220 BC Greeks (Eratosthenes of Cyrene)
- 1730 Euler $2^{2^5}+1=641\cdot 6700417$
- 1750–1800 Fermat, Gauss (Sieves - Tables)
- 1880 Landry & Le Lasseur:

$$2^{2^6}+1=274177\times 67280421310721$$

- 1919 Pierre and Eugène Carissan (Factoring Machine)
- 1970 Morrison & Brillhart

$$2^{2^7}+1=59649589127497217\times 5704689200685129054721$$

- 1982 Quadratic Sieve **QS** (Pomerance) $\rightsquigarrow$ Number Fields Sieve **NFS**
- 1987 Elliptic curves factoring **ECF** (Lenstra)

History of the "Art of Factoring"

220 BC Greeks (Eratosthenes of Cyrene)

History of the "Art of Factoring"

1730 Euler $2^{2^5}+1=641\cdot 6700417$

How did Euler factor $2^{2^5}+1$?

How did Euler factor $2^{2^5}+1$?

PROPOSITION *Suppose p is a prime factor of b^n+1. Then*

1. *p is a divisor of b^d+1 for some proper divisor d of n such that n/d is odd or*
2. *$p-1$ is divisible by $2n$.*

How did Euler factor $2^{2^5}+1$?

PROPOSITION *Suppose p is a prime factor of b^n+1. Then*

1. *p is a divisor of b^d+1 for some proper divisor d of n such that n/d is odd or*
2. *$p-1$ is divisible by $2n$.*

Application: Let $b=2$ and $n=2^5=64$. Then $2^{2^5}+1$ is prime or it is divisible by a prime p such that $p-1$ is divisible by 128.

How did Euler factor $2^{2^5}+1$?

PROPOSITION *Suppose p is a prime factor of b^n+1. Then*

1. *p is a divisor of b^d+1 for some proper divisor d of n such that n/d is odd or*
2. *$p-1$ is divisible by $2n$.*

Application: Let $b=2$ and $n=2^5=64$. Then $2^{2^5}+1$ is prime or it is divisible by a prime p such that $p-1$ is divisible by 128.

Note that
$1+1\times 128=3\times 43$, $1+2\times 128=257$ is prime,
$1+3\times 128=5\times 7\times 11$, $1+4\times 128=3^3\times 19$ and $1+5\cdot 128=641$ is prime.

Finally

$$\frac{2^{2^5}+1}{641}=\frac{4294967297}{641}=6700417$$

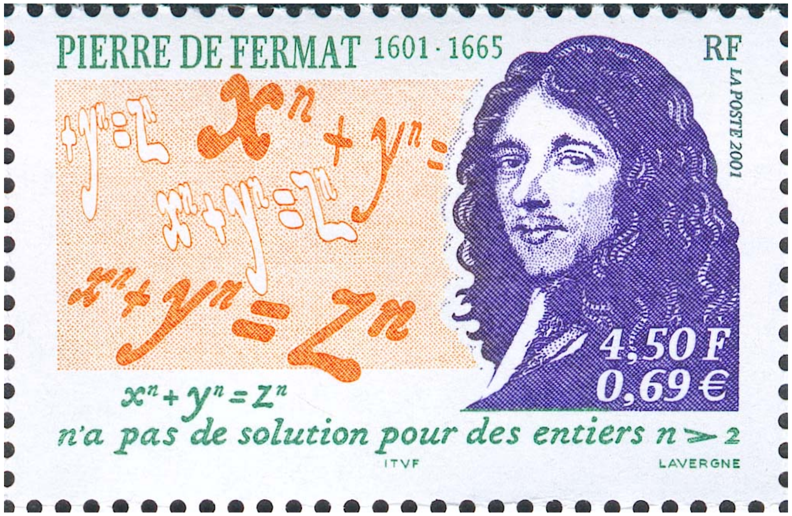
History of the "Art of Factoring"

1730 Euler $2^{2^5} + 1 = 641 \cdot 6700417$

History of the "Art of Factoring"

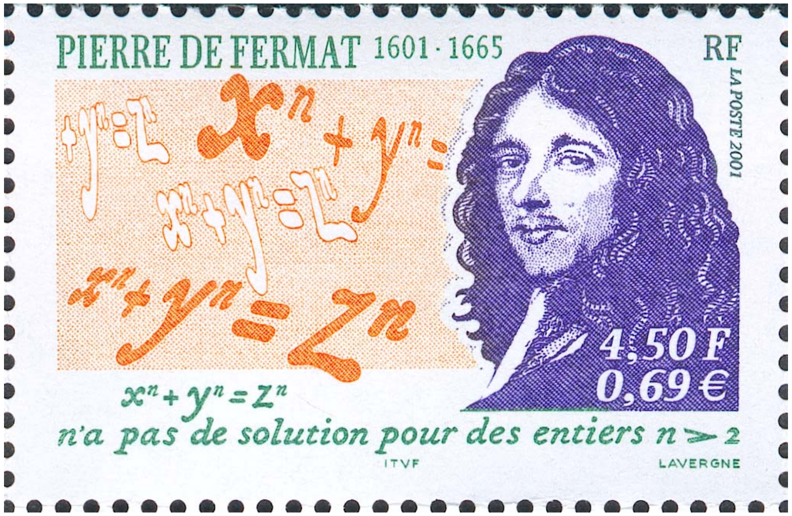
1750–1800 Fermat, Gauss (Sieves - Tables)

History of the "Art of Factoring"



1750–1800 Fermat, Gauss (Sieves - Tables)

History of the "Art of Factoring"



1750–1800 Fermat, Gauss (Sieves - Tables)

Factoring with sieves $N = x^2 - y^2 = (x-y)(x+y)$

Carissan's ancient Factoring Machine

Carissan's ancient Factoring Machine

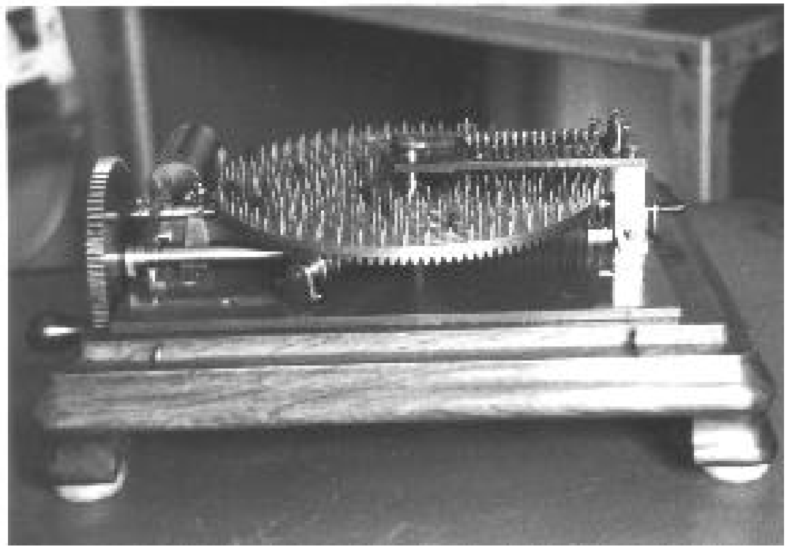


Figure 1: Conservatoire Nationale des Arts et Métiers in Paris

Carissan's ancient Factoring Machine

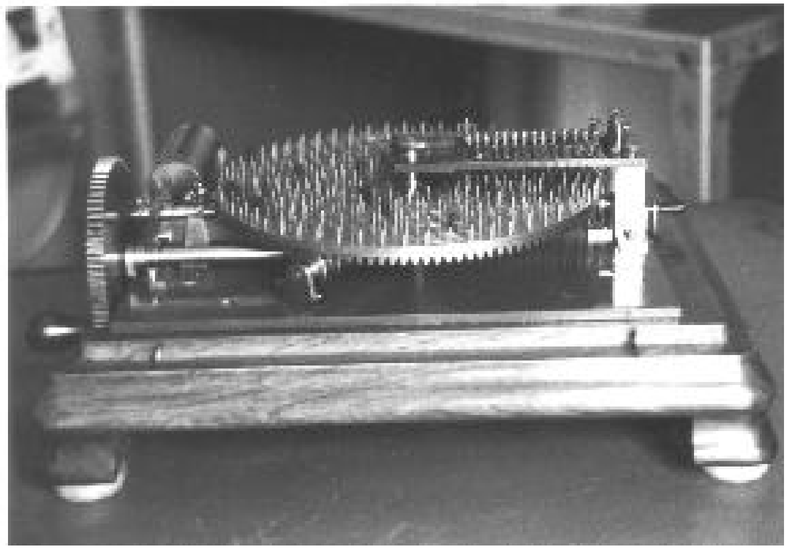


Figure 1: Conservatoire Nationale des Arts et Métiers in Paris

http://www.math.uwaterloo.ca/ shallit/Papers/carissan.html

Figure 2: Lieutenant Eugène Carissan

Figure 2: Lieutenant Eugène Carissan

$$225058681 = 229 \times 982789 \qquad \text{2 minutes}$$
$$3450315521 = 1409 \times 2418769 \qquad \text{3 minutes}$$
$$3570537526921 = 841249 \times 4244329 \qquad \text{18 minutes}$$

State of the "Art of Factoring"

1970 - John Brillhart & Michael A. Morrison

$$2^{2^7}+1 = 59649589127497217 \times 5704689200685129054721$$

State of the "Art of Factoring"

$$F_n = 2^{(2^n)} + 1$$

is called the n–th Fermat number

$F_{11} = 2^{2048} + 1 =$ 32,317,006,071,311,007,300,714,876,688,669,951,960,444,102,669,715,484,032,130,345,427,524,655,138,867,890,893,197,201,411,522,913,463,688,717,
960,921,898,019,494,119,559,150,490,921,095,088,152,386,448,283,120,630,877,367,300,996,091,750,197,750,389,652,106,796,057,638,384,067,
568,276,792,218,642,619,756,161,838,094,338,476,170,470,581,645,852,036,305,042,887,575,891,541,065,808,607,552,399,123,930,385,521,914,
333,389,668,342,420,684,974,786,564,569,494,856,176,035,326,322,058,077,805,659,331,026,192,708,460,314,150,258,592,864,177,116,725,943,
603,718,461,857,357,598,351,152,301,645,904,403,697,613,233,287,231,227,125,684,710,820,209,725,157,101,726,931,323,469,678,542,580,656,
697,935,045,997,268,352,998,638,215,525,166,389,437,335,543,602,135,433,229,604,645,318,478,604,952,148,193,555,853,611,059,596,230,657

= 319,489 × 974,849 × 167,988,556,341,760,475,137 × 3,560,841,906,445,833,920,513 ×
173,462,447,179,147,555,430,258,970,864,309,778,377,421,844,723,664,084,649,347,019,061,363,579,192,879,108,857,591,038,330,408,837,177,983,810,868,451,
546,421,940,712,978,306,134,189,864,280,826,014,542,758,708,589,243,873,685,563,973,118,948,869,399,158,545,506,611,147,420,216,132,557,017,260,564,139,
394,366,945,793,220,968,665,108,959,685,482,705,388,072,645,828,554,151,936,401,912,464,931,182,546,092,879,815,733,057,795,573,358,504,982,279,280,090,
942,872,567,591,518,912,118,622,751,714,319,229,788,100,979,251,036,035,496,917,279,912,663,527,358,783,236,647,193,154,777,091,427,745,377,038,294,
584,918,917,590,325,110,939,381,322,486,044,298,573,971,650,711,059,244,462,177,542,540,706,913,047,034,664,643,603,491,382,441,723,306,598,834,177

State of the "Art of Factoring"

$$F_n = 2^{(2^n)} + 1$$

is called the n–th Fermat number

$F_{11} = 2^{2048} + 1 =$ 32,317,006,071,311,007,300,714,876,688,669,951,960,444,102,669,715,484,032,130,345,427,524,655,138,867,890,893,197,201,411,522,913,463,688,717,
960,921,898,019,494,119,559,150,490,921,095,088,152,386,448,283,120,630,877,367,300,996,091,750,197,750,389,652,106,796,057,638,384,067,
568,276,792,218,642,619,756,161,838,094,338,476,170,470,581,645,852,036,305,042,887,575,891,541,065,808,607,552,399,123,930,385,521,914,
333,389,668,342,420,684,974,786,564,569,494,856,176,035,326,322,058,077,805,659,331,026,192,708,460,314,150,258,592,864,177,116,725,943,
603,718,461,857,357,598,351,152,301,645,904,403,697,613,233,287,231,227,125,684,710,820,209,725,157,101,726,931,323,469,678,542,580,656,
697,935,045,997,268,352,998,638,215,525,166,389,437,335,543,602,135,433,229,604,645,318,478,604,952,148,193,555,853,611,059,596,230,657

= 319,489 × 974,849 × 167,988,556,341,760,475,137 × 3,560,841,906,445,833,920,513 ×
173,462,447,179,147,555,430,258,970,864,309,778,377,421,844,723,664,084,649,347,019,061,363,579,192,879,108,857,591,038,330,408,837,177,983,810,868,451,
546,421,940,712,978,306,134,189,864,280,826,014,542,758,708,589,243,873,685,563,973,118,948,869,399,158,545,506,611,147,420,216,132,557,017,260,564,139,
394,366,945,793,220,968,665,108,959,685,482,705,388,072,645,828,554,151,936,401,912,464,931,182,546,092,879,815,733,057,795,573,358,504,982,279,280,090,
942,872,567,591,518,912,118,622,751,714,319,229,788,100,979,251,036,035,496,917,279,912,663,527,358,783,236,647,193,154,777,091,427,745,377,038,294,
584,918,917,590,325,110,939,381,322,486,044,298,573,971,650,711,059,244,462,177,542,540,706,913,047,034,664,643,603,491,382,441,723,306,598,834,177

Up to today only from F_0 to F_{11} are factores.

It is not known the factorization of

$$F_{12} = 2^{2^{12}} + 1$$

State of the "Art of Factoring"

$$F_n = 2^{(2^n)} + 1$$

is called the n–th Fermat number

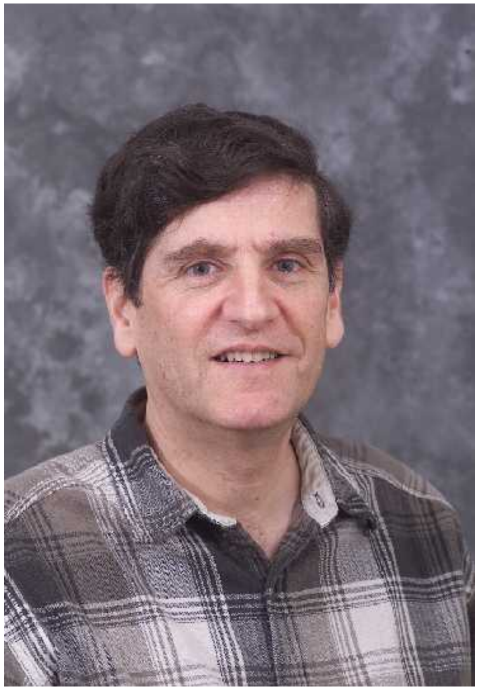
$F_{11} = 2^{2048} + 1 =$ 32,317,006,071,311,007,300,714,876,688,669,951,960,444,102,669,715,484,032,130,345,427,524,655,138,867,890,893,197,201,411,522,913,463,688,717,
960,921,898,019,494,119,559,150,490,921,095,088,152,386,448,283,120,630,877,367,300,996,091,750,197,750,389,652,106,796,057,638,384,067,
568,276,792,218,642,619,756,161,838,094,338,476,170,470,581,645,852,036,305,042,887,575,891,541,065,808,607,552,399,123,930,385,521,914,
333,389,668,342,420,684,974,786,564,569,494,856,176,035,326,322,058,077,805,659,331,026,192,708,460,314,150,258,592,864,177,116,725,943,
603,718,461,857,357,598,351,152,301,645,904,403,697,613,233,287,231,227,125,684,710,820,209,725,157,101,726,931,323,469,678,542,580,656,
697,935,045,997,268,352,998,638,215,525,166,389,437,335,543,602,135,433,229,604,645,318,478,604,952,148,193,555,853,611,059,596,230,657
= 319,489 × 974,849 × 167,988,556,341,760,475,137 × 3,560,841,906,445,833,920,513 ×
173,462,447,179,147,555,430,258,970,864,309,778,377,421,844,723,664,084,649,347,019,061,363,579,192,879,108,857,591,038,330,408,837,177,983,810,868,451,
546,421,940,712,978,306,134,189,864,280,826,014,542,758,708,589,243,873,685,563,973,118,948,869,399,158,545,506,611,147,420,216,132,557,017,260,564,139,
394,366,945,793,220,968,665,108,959,685,482,705,388,072,645,828,554,151,936,401,912,464,931,182,546,092,879,815,733,057,795,573,358,504,982,279,280,090,
942,872,567,591,518,912,118,622,751,714,319,229,788,100,979,251,036,035,496,917,279,912,663,527,358,783,236,647,193,154,777,091,427,745,377,038,294,
584,918,917,590,325,110,939,381,322,486,044,298,573,971,650,711,059,244,462,177,542,540,706,913,047,034,664,643,603,491,382,441,723,306,598,834,177

Up to today only from F_0 to F_{11} are factores.

It is not known the factorization of

$$F_{12} = 2^{2^{12}} + 1$$

State of the "Art of Factoring"



1982 - Carl Pomerance - Quadratic Sieve

State of the "Art of Factoring"

1987 - Hendrik Lenstra - Elliptic curves factoring

Contemporary Factoring

Contemporary Factoring

❶ 1994, Quadratic Sieve (QS): (8 months, 600 volunteers, 20 nations)

D.Atkins, M. Graff, A. Lenstra, P. Leyland

$$RSA_{129} = 114381625757888867669235779976146612010218296721242362562561842935706$$
$$935245733897830597123563958705058989075147599290026879543541 =$$
$$= 3490529510847650949147849619903898133417764638493387843990820577 \times$$
$$32769132993266709549961988190834461413177642967992942539798288533$$

Contemporary Factoring

❶ 1994, Quadratic Sieve (QS): (8 months, 600 volunteers, 20 nations)

D.Atkins, M. Graff, A. Lenstra, P. Leyland

$$RSA_{129} = 114381625757888867669235779976146612010218296721242362562561842935706\\935245733897830597123563958705058989075147599290026879543541 =\\= 3490529510847650949147849619903898133417764638493387843990820577\times\\32769132993266709549961988190834461413177642967992942539798288533$$

❷ (February 2 1999), Number Field Sieve (NFS): (160 Sun, 4 months)

$$RSA_{155} = 109417386415705274218097073220403576120037329454492059909138421314763499842\\88934784717997257891267332497625752899781833797076537244027146743531593354333897 =\\= 102639592829741105772054196573991675900716567808038066803341933521790711307779\times\\106603488380168454820927220360012878679207958575989291522270608237193062808643$$

Contemporary Factoring

❶ 1994, Quadratic Sieve (QS): (8 months, 600 volunteers, 20 nations)

D.Atkins, M. Graff, A. Lenstra, P. Leyland

$$RSA_{129} = 114381625757888867669235779976146612010218296721242362562561842935706$$
$$935245733897830597123563958705058989075147599290026879543541 =$$
$$= 3490529510847650949147849619903898133417764638493387843990820577 \times$$
$$32769132993266709549961988190834461413177642967992942539798288533$$

❷ (February 2 1999), Number Field Sieve (NFS): (160 Sun, 4 months)

$$RSA_{155} = 109417386415705274218097073220403576120037329454492059909138421314763499842$$
$$88934784717997257891267332497625752899781833797076537244027146743531593354333897 =$$
$$= 102639592829741105772054196573991675900716567808038066803341933521790711307779 \times$$
$$106603488380168454820927220360012878679207958575989291522270608237193062808643$$

❸ (December 3, 2003) (NFS): J. Franke et al. (174 decimal digits)

$$RSA_{576} = 1881988129206079638386972394616504398071635633794173827007633564229888597152346$$
$$65485319060606504743045317388011303396716199692321205734031879550656996221305168759307650257059 =$$
$$= 398075086424064937397125500550386491199064362342526708406385189575946388957261768583317 \times$$
$$472772146107435302536223071973048224632914695302097116459852171130520711256363590397527$$

Contemporary Factoring

1. 1994, Quadratic Sieve (QS): (8 months, 600 volunteers, 20 nations)
   D.Atkins, M. Graff, A. Lenstra, P. Leyland

$$RSA_{129} = 114381625757888867669235779976146612010218296721242362562561842935706$$
$$935245733897830597123563958705058989075147599290026879543541 =$$
$$= 3490529510847650949147849619903898133417764638493387843990820577 \times$$
$$32769132993266709549961988190834461413177642967992942539798288533$$

2. (February 2 1999), Number Field Sieve (NFS): (160 Sun, 4 months)

$$RSA_{155} = 109417386415705274218097073220403576120037329454492059909138421314763499842$$
$$88934784717997257891267332497625752899781833797076537244027146743531593354333897 =$$
$$= 102639592829741105772054196573991675900716567808038066803341933521790711307779 \times$$
$$106603488380168454820927220360012878679207958575989291522270608237193062808643$$

3. (December 3, 2003) (NFS): J. Franke et al. (174 decimal digits)

$$RSA_{576} = 188198812920607963838697239461650439807163563379417382700763356422988859715234$$
$$6654853190606065047430453173880113033967161996923212057340318795506569962213051687593076502570 59 =$$
$$= 398075086424064937397125500550386491199064362342526708406385189575946388957261768583317 \times$$
$$472772146107435302536223071973048224632914695302097116459852171130520711256363590397527$$

4. Elliptic curves factoring: introduced by H. Lenstra. suitable to detect small factors (50 digits)

Contemporary Factoring

1. 1994, Quadratic Sieve (QS): (8 months, 600 volunteers, 20 nations) D.Atkins, M. Graff, A. Lenstra, P. Leyland

$$RSA_{129} = 114381625757888867669235779976146612010218296721242362562561842935706$$
$$935245733897830597123563958705058989075147599290026879543541 =$$
$$= 3490529510847650949147849619903898133417764638493387843990820577 \times$$
$$32769132993266709549961988190834461413177642967992942539798288533$$

2. (February 2 1999), Number Field Sieve (NFS): (160 Sun, 4 months)

$$RSA_{155} = 109417386415705274218097073220403576120037329454492059909138421314763499842$$
$$88934784717997257891267332497625752899781833797076537244027146743531593354333897 =$$
$$= 102639592829741105772054196573991675900716567808038066803341933521790711307779 \times$$
$$106603488380168454820927220360012878679207958575989291522270608237193062808643$$

3. (December 3, 2003) (NFS): J. Franke et al. (174 decimal digits)

$$RSA_{576} = 188198812920607963838697239461650439807163563379417382700763356422988859715234$$
$$665485319060606504743045317388011303396716199692321205734031879550656996221305168759307650257059 =$$
$$= 398075086424064937397125500550386491199064362342526708406385189575946388957261768583317 \times$$
$$472772146107435302536223071973048224632914695302097116459852171130520711256363590397527$$

4. Elliptic curves factoring: introduced by H. Lenstra. suitable to detect small factors (50 digits)

all have "sub–exponential complexity"

The factorization of RSA_{200}

$RSA_{200} =$ 2799783391122132787082946763872260162107044678695542853756000992932612840010
7609345671052955360856061822351910951365788637105954482006576775098580557613
579098734950144178863178946295187237869221823983

The factorization of RSA_{200}

RSA_{200} = 2799783391122132787082946763872260162107044678695542853756000992932612840010
7609345671052955360856061822351910951365788637105954482006576775098580557613
579098734950144178863178946295187237869221823983

Date: Mon, 9 May 2005 18:05:10 +0200 (CEST) From: "Thorsten Kleinjung" Subject: rsa200

We have factored RSA200 by GNFS. The factors are

35324619344027701212726049781984643686711974001976 25023649303468776121253679423200058547956528088349

and

79258699544783330333470858414800596877379758573642 19960734330341455767872818152135381409304740185467

We did lattice sieving for most special q between 3e8 and 11e8 using mainly factor base bounds of 3e8 on the algebraic side and 18e7 on the rational side. The bounds for large primes were 2^{35}. This produced 26e8 relations. Together with 5e7 relations from line sieving the total yield was 27e8 relations. After removing duplicates 226e7 relations remained. A filter job produced a matrix with 64e6 rows and columns, having 11e9 non-zero entries. This was solved by Block-Wiedemann.

Sieving has been done on a variety of machines. We estimate that lattice sieving would have taken 55 years on a single 2.2 GHz Opteron CPU. Note that this number could have been improved if instead of the PIII- binary which we used for sieving, we had used a version of the lattice-siever optimized for Opteron CPU's which we developed in the meantime. The matrix step was performed on a cluster of 80 2.2 GHz Opterons connected via a Gigabit network and took about 3 months.

We started sieving shortly before Christmas 2003 and continued until October 2004. The matrix step began in December 2004. Line sieving was done by P. Montgomery and H. te Riele at the CWI, by F. Bahr and his family.

More details will be given later.

F. Bahr, M. Boehm, J. Franke, T. Kleinjung

Factorization of RSA_{768}

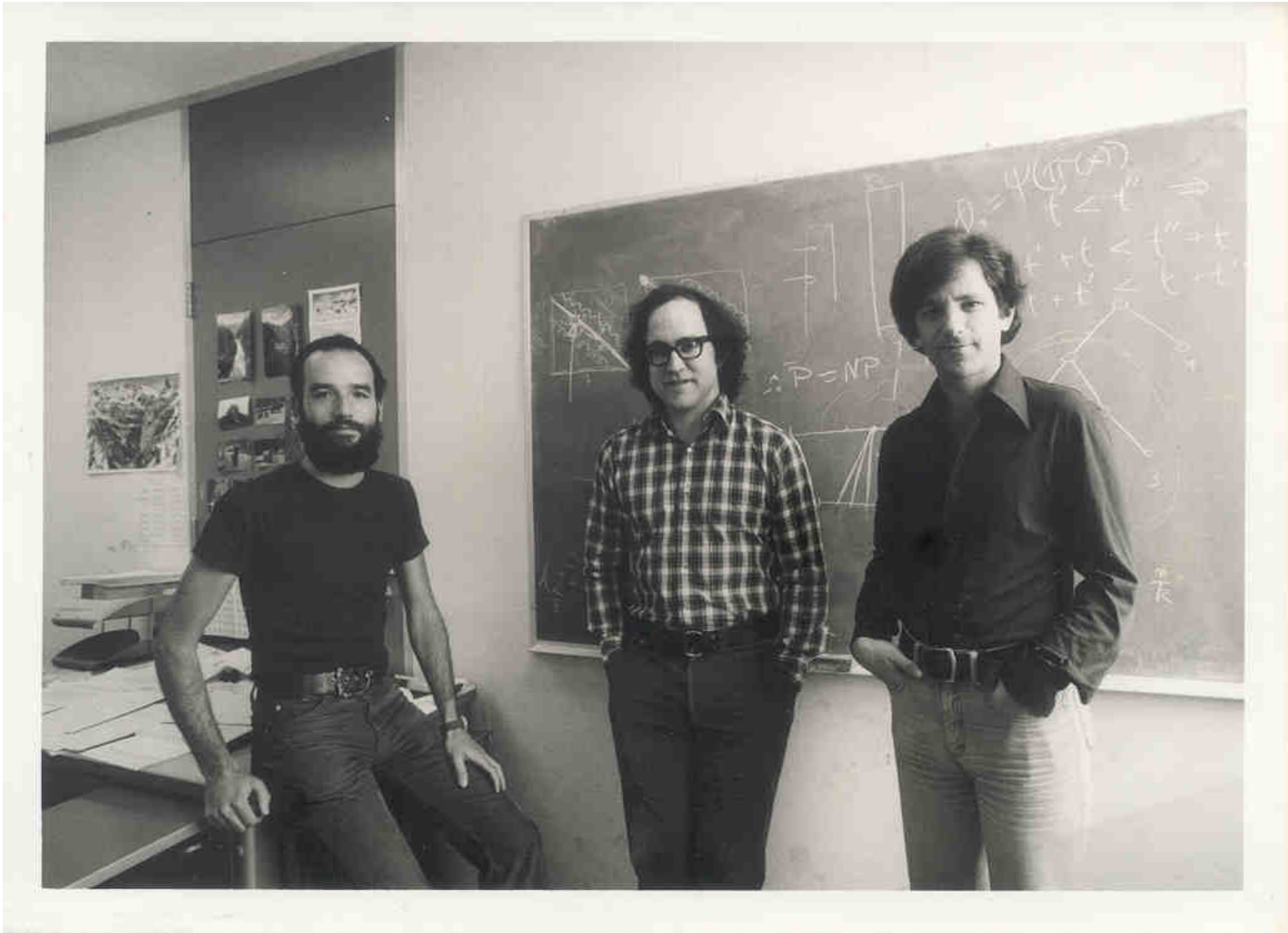
RSA-768 [edit]

RSA-768 has 232 decimal digits (768 bits), and was factored on December 12, 2009 by Thorsten Kleinjung, Kazumaro Aoki, Jens Franke, Arjen K. Lenstra, Emmanuel Thomé, Pierrick Gaudry, Alexander Kruppa, Peter Montgomery, Joppe W. Bos, Dag Arne Osvik, Herman te Riele, Andrey Timofeev, and Paul Zimmermann.[31]

```
RSA-768 = 12301866845301177551304949583849627207728535695953347921973224521517264005
          07263657518745202199786469389956474942774063845925192557326303453731548268
          50791702612214291346167042921431160222124047927473779408066535141959745985
          6902143413
```

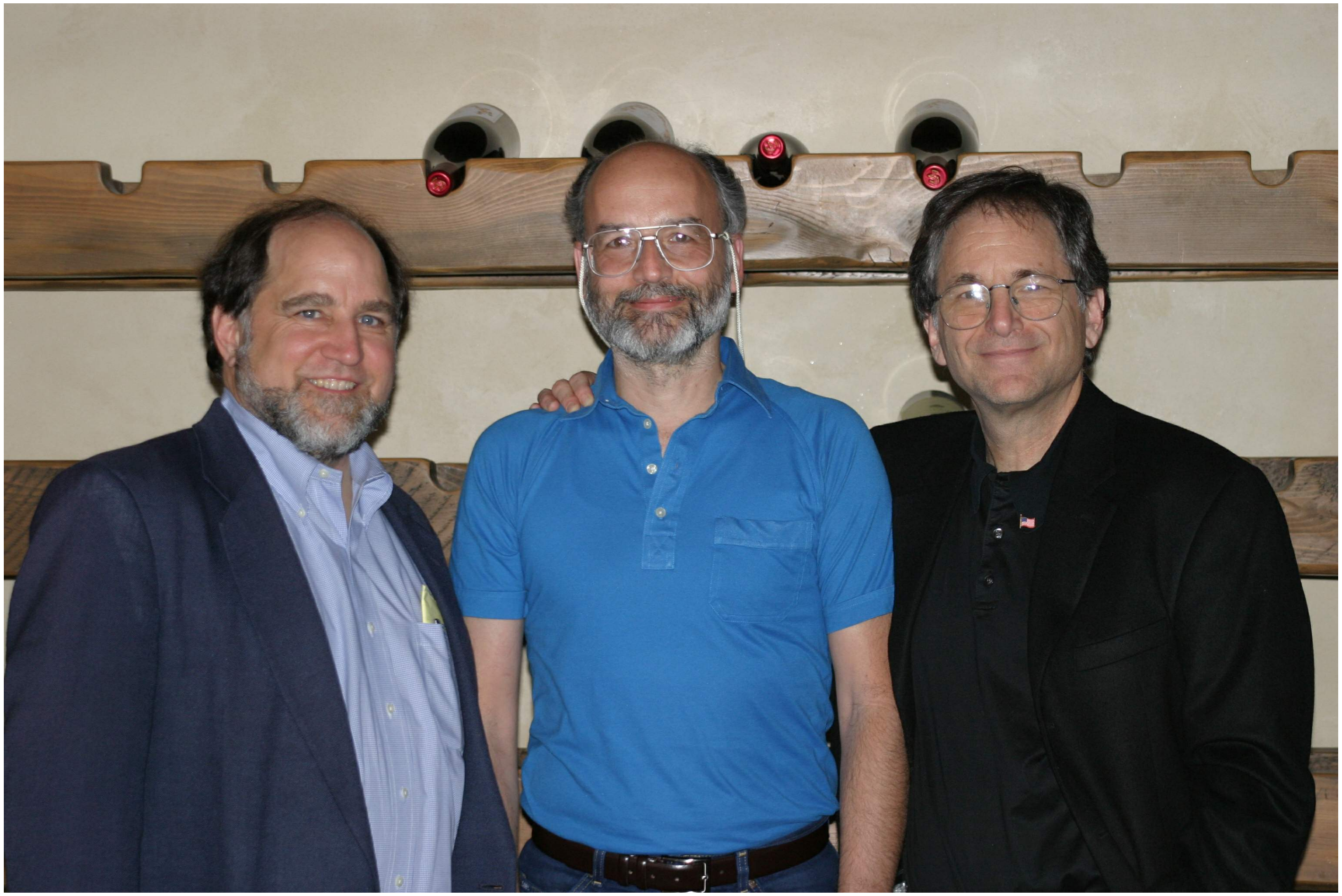
```
RSA-768 = 33478071698956898786044169848212690817704794983713768568912431388982883793
          878002287614711652531743087737814467999489
        × 36746043666799590428244633799627952632279158164343087642676032283815739666
          511279233373417143396810270092798736308917
```

RSA



Adi Shamir, Ron L. Rivest, Leonard Adleman (1978)

RSA



Ron L. Rivest, Adi Shamir, Leonard Adleman (2003)

The RSA cryptosystem

The RSA cryptosystem

1978 R. L. Rivest, A. Shamir, L. Adleman (Patent expired in 1998)

The RSA cryptosystem

1978 R. L. Rivest, A. Shamir, L. Adleman (Patent expired in 1998)

Problem: Alice wants to send the message $\mathcal{P}$ to Bob so that Charles cannot read it

The RSA cryptosystem

1978 R. L. Rivest, A. Shamir, L. Adleman (Patent expired in 1998)

Problem: Alice wants to send the message $\mathcal{P}$ to Bob so that Charles cannot read it

$$
\begin{array}{ccc}
A\ (Alice) & \longrightarrow & B\ (Bob) \\
 & \uparrow & \\
 & C\ (Charles) &
\end{array}
$$

The RSA cryptosystem

1978 R. L. Rivest, A. Shamir, L. Adleman (Patent expired in 1998)

Problem: Alice wants to send the message $\mathcal{P}$ to Bob so that Charles cannot read it

$$\boxed{\begin{array}{ccc} A\ (Alice) & \longrightarrow & B\ (Bob) \\ & \uparrow & \\ & C\ (Charles) & \end{array}}$$

❶

❷

❸

❹

The RSA cryptosystem

1978 R. L. Rivest, A. Shamir, L. Adleman (Patent expired in 1998)

Problem: Alice wants to send the message $\mathcal{P}$ to Bob so that Charles cannot read it

$$\begin{array}{ccc} A\ (Alice) & \longrightarrow & B\ (Bob) \\ & \uparrow & \\ & C\ (Charles) & \end{array}$$

1. Key generation — Bob has to do it
2.
3.
4.

The RSA cryptosystem

1978 R. L. Rivest, A. Shamir, L. Adleman (Patent expired in 1998)

Problem: Alice wants to send the message $\mathcal{P}$ to Bob so that Charles cannot read it

$$\begin{array}{ccc} A\ (Alice) & \longrightarrow & B\ (Bob) \\ & \uparrow & \\ & C\ (Charles) & \end{array}$$

1. Key generation — Bob has to do it
2. Encryption — Alice has to do it
3.
4.

The RSA cryptosystem

1978 R. L. Rivest, A. Shamir, L. Adleman (Patent expired in 1998)

Problem: Alice wants to send the message $\mathcal{P}$ to Bob so that Charles cannot read it

$$\begin{array}{ccc} A\ (Alice) & \longrightarrow & B\ (Bob) \\ & \uparrow & \\ & C\ (Charles) & \end{array}$$

1. Key generation — Bob has to do it
2. Encryption — Alice has to do it
3. Decryption — Bob has to do it
4.

The RSA cryptosystem

1978 R. L. Rivest, A. Shamir, L. Adleman (Patent expired in 1998)

Problem: Alice wants to send the message $\mathcal{P}$ to Bob so that Charles cannot read it

$$\begin{array}{ccc} A\ (Alice) & \longrightarrow & B\ (Bob) \\ & \uparrow & \\ & C\ (Charles) & \end{array}$$

1. KEY GENERATION — Bob has to do it
2. ENCRYPTION — Alice has to do it
3. DECRYPTION — Bob has to do it
4. ATTACK — Charles would like to do it

Bob: Key generation

Bob: Key generation

✍

✍

✍

✍

✍

Bob: Key generation

- He chooses randomly p and q primes $(p, q \approx 10^{100})$
-
-
-
-

Bob: Key generation

- He chooses randomly p and q primes $(p, q \approx 10^{100})$
- He computes $M = p \times q$, $\varphi(M) = (p-1) \times (q-1)$
-
-
-

Bob: Key generation

- ✍ He chooses randomly p and q primes $(p, q \approx 10^{100})$
- ✍ He computes $M = p \times q$, $\varphi(M) = (p-1) \times (q-1)$
- ✍ He chooses an integer e s.t.
- ✍
- ✍

Bob: Key generation

- He chooses randomly p and q primes $(p, q \approx 10^{100})$
- He computes $M = p \times q$, $\varphi(M) = (p-1) \times (q-1)$
- He chooses an integer e s.t.

$$0 \leq e \leq \varphi(M) \quad \text{and} \quad \gcd(e, \varphi(M)) = 1$$

Bob: Key generation

- He chooses randomly p and q primes $(p, q \approx 10^{100})$
- He computes $M = p \times q$, $\varphi(M) = (p-1) \times (q-1)$
- He chooses an integer e s.t.

$$0 \le e \le \varphi(M) \quad \text{and} \quad \gcd(e, \varphi(M)) = 1$$

NOTE. One could take $e = 3$ and $p \equiv q \equiv 2 \bmod 3$

-
-

Bob: Key generation

- ✍ He chooses randomly p and q primes $(p, q \approx 10^{100})$
- ✍ He computes $M = p \times q$, $\varphi(M) = (p-1) \times (q-1)$
- ✍ He chooses an integer e s.t.

$$0 \leq e \leq \varphi(M) \text{ and } \gcd(e, \varphi(M)) = 1$$

NOTE. One could take $e = 3$ and $p \equiv q \equiv 2 \bmod 3$

Experts recommend $e = 2^{16} + 1$

- ✍
- ✍

Bob: Key generation

- He chooses randomly p and q primes $(p, q \approx 10^{100})$
- He computes $M = p \times q$, $\varphi(M) = (p-1) \times (q-1)$
- He chooses an integer e s.t.

$$0 \leq e \leq \varphi(M) \quad \text{and} \quad \gcd(e, \varphi(M)) = 1$$

NOTE. One could take $e = 3$ and $p \equiv q \equiv 2 \bmod 3$

Experts recommend $e = 2^{16} + 1$

- He computes arithmetic inverse d of e modulo $\varphi(M)$
-

Bob: Key generation

- ✍ He chooses randomly p and q primes $(p, q \approx 10^{100})$
- ✍ He computes $M = p \times q$, $\varphi(M) = (p-1) \times (q-1)$
- ✍ He chooses an integer e s.t.

  $$0 \leq e \leq \varphi(M) \quad \text{and} \quad \gcd(e, \varphi(M)) = 1$$

  NOTE. One could take $e = 3$ and $p \equiv q \equiv 2 \bmod 3$

  Experts recommend $e = 2^{16} + 1$
- ✍ He computes arithmetic inverse d of e modulo $\varphi(M)$

  (i.e. $d \in \mathbb{N}$ (unique $\leq \varphi(M)$) s.t. $e \times d \equiv 1 \pmod{\varphi(M)}$)
- ✍

Bob: Key generation

- ✍ He chooses randomly p and q primes $(p, q \approx 10^{100})$
- ✍ He computes $M = p \times q$, $\varphi(M) = (p-1) \times (q-1)$
- ✍ He chooses an integer e s.t.

  $$0 \le e \le \varphi(M) \text{ and } \gcd(e, \varphi(M)) = 1$$

  NOTE. One could take $e = 3$ and $p \equiv q \equiv 2 \bmod 3$

  Experts recommend $e = 2^{16} + 1$
- ✍ He computes arithmetic inverse d of e modulo $\varphi(M)$

  (i.e. $d \in \mathbb{N}$ (unique $\le \varphi(M)$) s.t. $e \times d \equiv 1 \pmod{\varphi(M)}$)
- ✍ Publishes (M, e) **public key** and hides **secret key** d

Bob: Key generation

- ✍ He chooses randomly p and q primes $(p, q \approx 10^{100})$
- ✍ He computes $M = p \times q$, $\varphi(M) = (p-1) \times (q-1)$
- ✍ He chooses an integer e s.t.

$$0 \leq e \leq \varphi(M) \quad \text{and} \quad \gcd(e, \varphi(M)) = 1$$

NOTE. One could take $e = 3$ and $p \equiv q \equiv 2 \bmod 3$

Experts recommend $e = 2^{16} + 1$

- ✍ He computes arithmetic inverse d of e modulo $\varphi(M)$

(i.e. $d \in \mathbb{N}$ (unique $\leq \varphi(M)$) s.t. $e \times d \equiv 1 \pmod{\varphi(M)}$)

- ✍ Publishes (M, e) **public key** and hides **secret key** d

Problem: How does Bob do all this?- We will go came back to it!

Alice: Encryption

Alice: Encryption

Represent the message $\mathcal{P}$ as an element of $\mathbb{Z}/M\mathbb{Z}$

Alice: Encryption

Represent the message $\mathcal{P}$ as an element of $\mathbb{Z}/M\mathbb{Z}$

(for example) $\boxed{A \leftrightarrow 1 \quad B \leftrightarrow 2 \quad C \leftrightarrow 3 \quad \ldots \quad Z \leftrightarrow 26 \quad AA \leftrightarrow 27 \ldots}$

Alice: Encryption

Represent the message $\mathcal{P}$ as an element of $\mathbb{Z}/M\mathbb{Z}$

(for example) $\boxed{A \leftrightarrow 1 \quad B \leftrightarrow 2 \quad C \leftrightarrow 3 \quad \ldots \quad Z \leftrightarrow 26 \quad AA \leftrightarrow 27\ldots}$

`Sukumar` $\leftrightarrow 19 \cdot 26^6 + 21 \cdot 26^5 + 11 \cdot 26^4 + 21 \cdot 26^3 + 12 \cdot 26^2 + 1 \cdot 26 + 18 = 6124312628$

Note. Better if texts are not too short. Otherwise one performs some *padding*

Alice: Encryption

Represent the message $\mathcal{P}$ as an element of $\mathbb{Z}/M\mathbb{Z}$

(for example) $\boxed{A \leftrightarrow 1 \quad B \leftrightarrow 2 \quad C \leftrightarrow 3 \quad \ldots \quad Z \leftrightarrow 26 \quad AA \leftrightarrow 27 \ldots}$

`Sukumar` $\leftrightarrow 19 \cdot 26^6 + 21 \cdot 26^5 + 11 \cdot 26^4 + 21 \cdot 26^3 + 12 \cdot 26^2 + 1 \cdot 26 + 18 = 6124312628$

Note. Better if texts are not too short. Otherwise one performs some *padding*

$$\boxed{\mathcal{C} = E(\mathcal{P}) = \mathcal{P}^e \pmod{M}}$$

Alice: Encryption

Represent the message $\mathcal{P}$ as an element of $\mathbb{Z}/M\mathbb{Z}$

(for example) $\boxed{A \leftrightarrow 1 \quad B \leftrightarrow 2 \quad C \leftrightarrow 3 \quad \ldots \quad Z \leftrightarrow 26 \quad AA \leftrightarrow 27 \ldots}$

$$\texttt{Sukumar} \leftrightarrow 19 \cdot 26^6 + 21 \cdot 26^5 + 11 \cdot 26^4 + 21 \cdot 26^3 + 12 \cdot 26^2 + 1 \cdot 26 + 18 = 6124312628$$

Note. Better if texts are not too short. Otherwise one performs some *padding*

$$\boxed{\mathcal{C} = E(\mathcal{P}) = \mathcal{P}^e \pmod{M}}$$

Example: $p = 9049465727$, $q = 8789181607$, $M = 79537397720925283289$, $e = 2^{16} + 1 = 65537$, $\mathcal{P} = \texttt{Sukumar}$:

Alice: Encryption

Represent the message $\mathcal{P}$ as an element of $\mathbb{Z}/M\mathbb{Z}$

(for example) $\boxed{A \leftrightarrow 1 \quad B \leftrightarrow 2 \quad C \leftrightarrow 3 \quad \ldots \quad Z \leftrightarrow 26 \quad AA \leftrightarrow 27 \ldots}$

$$\texttt{Sukumar} \leftrightarrow 19 \cdot 26^6 + 21 \cdot 26^5 + 11 \cdot 26^4 + 21 \cdot 26^3 + 12 \cdot 26^2 + 1 \cdot 26 + 18 = 6124312628$$

Note. Better if texts are not too short. Otherwise one performs some *padding*

$$\boxed{\mathcal{C} = E(\mathcal{P}) = \mathcal{P}^e \pmod{M}}$$

Example: $p = 9049465727$, $q = 8789181607$, $M = 79537397720925283289$, $e = 2^{16} + 1 = 65537$, $\mathcal{P} = \texttt{Sukumar}$:

$$E(\texttt{Sukumar}) = 6124312628^{65537} \pmod{79537397720925283289}$$

$$= 25439695120356558116 = \mathcal{C} = \texttt{JGEBNBAUYTCOFJ}$$

Bob: Decryption

Bob: Decryption

$$\boxed{\mathcal{P} = D(\mathcal{C}) = \mathcal{C}^d \pmod{M}}$$

Bob: Decryption

$$\boxed{\mathcal{P} = D(\mathcal{C}) = \mathcal{C}^d \pmod{M}}$$

Note. Bob decrypts because he is the only one that knows d.

Bob: Decryption

$$\mathcal{P} = D(\mathcal{C}) = \mathcal{C}^d \pmod{M}$$

Note. Bob decrypts because he is the only one that knows d.

Theorem. (Euler) If $a, m \in \mathbb{N}$, $\gcd(a,m) = 1$,

$$a^{\varphi(m)} \equiv 1 \pmod{m}.$$

If $n_1 \equiv n_2 \bmod \varphi(m)$ then $a^{n_1} \equiv a^{n_2} \bmod m$.

Bob: Decryption

$$\boxed{\mathcal{P} = D(\mathcal{C}) = \mathcal{C}^d \pmod{M}}$$

Note. Bob decrypts because he is the only one that knows d.

Theorem. (Euler) If $a, m \in \mathbb{N}$, $\gcd(a, m) = 1$,

$$a^{\varphi(m)} \equiv 1 \pmod{m}.$$

If $n_1 \equiv n_2 \bmod \varphi(m)$ then $a^{n_1} \equiv a^{n_2} \bmod m$.

Therefore $(ed \equiv 1 \bmod \varphi(M))$

$$\boxed{D(E(\mathcal{P})) = \mathcal{P}^{ed} \equiv \mathcal{P} \bmod M}$$

Bob: Decryption

$$\boxed{\mathcal{P} = D(\mathcal{C}) = \mathcal{C}^d \pmod{M}}$$

Note. Bob decrypts because he is the only one that knows d.

Theorem. (Euler) If $a, m \in \mathbb{N}$, $\gcd(a, m) = 1$,

$$a^{\varphi(m)} \equiv 1 \pmod{m}.$$

If $n_1 \equiv n_2 \bmod \varphi(m)$ then $a^{n_1} \equiv a^{n_2} \bmod m$.

Therefore $(ed \equiv 1 \bmod \varphi(M))$

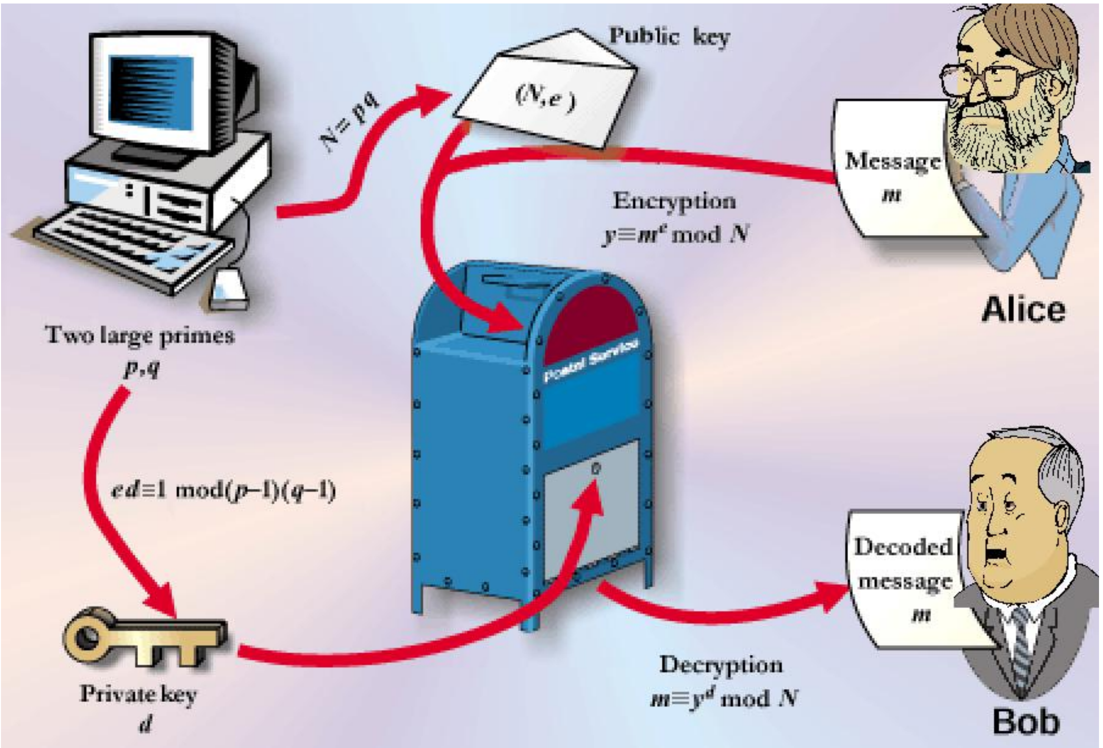
$$\boxed{D(E(\mathcal{P})) = \mathcal{P}^{ed} \equiv \mathcal{P} \bmod M}$$

Example(cont.):$d = 65537^{-1} \bmod \varphi(9049465727 \cdot 8789181607) = 57173914060643780153$

$D(\texttt{JGEBNBAUYTCOFJ}) =$

$2543969512035655811 6^{57173914060643780153} (\text{mod} 79537397720925283289) =$ `Sukumar`

RSA at work


Public key
(N,e)
$N=pq$
Message
m
Encryption
$y \equiv m^e \bmod N$
Alice
Two large primes
p,q
Postal Service
$ed \equiv 1 \bmod (p-1)(q-1)$
Decoded
message
m
Private key
d
Decryption
$m \equiv y^d \bmod N$
Bob

Repeated squaring algorithm

Repeated squaring algorithm

Problem: How does one compute $a^b \bmod c$?

Repeated squaring algorithm

Problem: How does one compute $a^b \bmod c$?

$$2543969512035655811 6^{57173914060643780153} \pmod{79537397720925283289}$$

Repeated squaring algorithm

Problem: How does one compute $a^b \bmod c$?

$$2543969512035655811 6^{5717391406064378015 3} \pmod{79537397720925283289}$$

✍

✍

✍

Repeated squaring algorithm

Problem: How does one compute $a^b \bmod c$?

$$2543969512035655811 6^{57173914060643780153} \pmod{79537397720925283289}$$

✍ Compute the binary expansion $b = \sum_{j=0}^{[\log_2 b]} \epsilon_j 2^j$

✍

✍

Repeated squaring algorithm

Problem: How does one compute $a^b \bmod c$?

$$2543969512035655811 6^{5717391406064378015 3} \pmod{7953739772092528328 9}$$

- Compute the binary expansion $b = \sum_{j=0}^{[\log_2 b]} \epsilon_j 2^j$

  5717391406064378015 3=110001100101110010100010111110101011110011011000100100011000111001

-

-

Repeated squaring algorithm

Problem: How does one compute $a^b \bmod c$?

$$254396951203565581 16^{5717391406064378015 3}(\mathrm{mod}\,79537397720925283289)$$

- Compute the binary expansion $b = \sum_{j=0}^{[\log_2 b]} \epsilon_j 2^j$

$5717391406064378015 3 = 110001100101110010100010111110101011110011011000100100011000111001$

- Compute recursively $a^{2^j} \bmod c, j = 1, \ldots, [\log_2 b]$:

-

Repeated squaring algorithm

Problem: How does one compute $a^b \bmod c$?

$$2543969512035655811 6^{5717391406064378015 3}(\bmod 7953739772092528328 9)$$

- Compute the binary expansion $b = \sum_{j=0}^{[\log_2 b]} \epsilon_j 2^j$

$5717391406064378015 3 = 110001100101110010100010111110101011110011011000100100011000111001$

- Compute recursively $a^{2^j} \bmod c, j = 1, \ldots, [\log_2 b]$:

$$a^{2^j} \bmod c = \left(a^{2^{j-1}} \bmod c\right)^2 \bmod c$$

-

Repeated squaring algorithm

Problem: How does one compute $a^b \bmod c$?

$$2543969512035655811 6^{5717391406064378 0153} (\bmod 7953739772092528 3289)$$

- Compute the binary expansion $b = \sum_{j=0}^{[\log_2 b]} \epsilon_j 2^j$

$5717391406064378 0153 = 110001100101110010100010111110101011110011011000100100011000111001$

- Compute recursively $a^{2^j} \bmod c, j = 1, \ldots, [\log_2 b]$:

$$a^{2^j} \bmod c = \left(a^{2^{j-1}} \bmod c\right)^2 \bmod c$$

- Multiply the $a^{2^j} \bmod c$ with $\epsilon_j = 1$

Repeated squaring algorithm

Problem: How does one compute $a^b \bmod c$?

$$2543969512035655811 6^{5717391406064378015 3}(\bmod 7953739772092528328 9)$$

- <u>Compute the binary expansion</u> $b = \sum_{j=0}^{[\log_2 b]} \epsilon_j 2^j$

571739140606437801 53=110001100101110010100010111110101011110011011000100100011000111001

- <u>Compute recursively</u> $a^{2^j} \bmod c, j = 1, \ldots, [\log_2 b]$:

$$a^{2^j} \bmod c = \left(a^{2^{j-1}} \bmod c\right)^2 \bmod c$$

- <u>Multiply</u> the $a^{2^j} \bmod c$ with $\epsilon_j = 1$

$$a^b \bmod c = \left(\prod_{j=0,\epsilon_j=1}^{[\log_2 b]} a^{2^j} \bmod c\right) \bmod c$$

$\#\{$**oper. in** $\mathbb{Z}/c\mathbb{Z}$ **to compute** $a^b \bmod c\} \leq 2\log_2 b$

$\#\{$**oper. in** $\mathbb{Z}/c\mathbb{Z}$ **to compute** $a^b \bmod c\} \leq 2\log_2 b$

`JGEBNBAUYTCOFJ` is decrypted with 131 operations in

$$\mathbb{Z}/79537397720925283289\mathbb{Z}$$

$\#\{$**oper. in** $\mathbb{Z}/c\mathbb{Z}$ **to compute** $a^b \bmod c\} \leq 2\log_2 b$

`JGEBNBAUYTCOFJ` is decrypted with 131 operations in

$$\mathbb{Z}/795373977209252832 89\mathbb{Z}$$

Pseudo code: $e_c(a,b) = a^b \bmod c$

$\#\{$**oper. in** $\mathbb{Z}/c\mathbb{Z}$ **to compute** $a^b \bmod c\} \le 2\log_2 b$

`JGEBNBAUYTCOFJ` is decrypted with 131 operations in

$$\mathbb{Z}/79537397720925283289\mathbb{Z}$$

PSEUDO CODE: $e_c(a,b) = a^b \bmod c$

$e_c(a,b)$	`=`	`if`	$b=1$	`then`	$a \bmod c$
		`if`	$2 \mid b$	`then`	$e_c(a,\frac{b}{2})^2 \bmod c$
		`else`			$a * e_c(a,\frac{b-1}{2})^2 \bmod c$

$\#\{$**oper. in** $\mathbb{Z}/c\mathbb{Z}$ **to compute** $a^b \bmod c\} \le 2\log_2 b$

`JGEBNBAUYTCOFJ` is decrypted with 131 operations in

$$\mathbb{Z}/795373977209252832 89\mathbb{Z}$$

PSEUDO CODE: $e_c(a,b) = a^b \bmod c$

$e_c(a,b)$	`=`	`if`	$b=1$	`then`	$a \bmod c$
		`if`	$2 \mid b$	`then`	$e_c(a, \frac{b}{2})^2 \bmod c$
		`else`			$a * e_c(a, \frac{b-1}{2})^2 \bmod c$

To encrypt with $e = 2^{16}+1$, only 17 operations in $\mathbb{Z}/M\mathbb{Z}$ are enough

Key generation

Key generation

Problem. Produce a random prime $p \approx 10^{100}$

Probabilistic algorithm (type Las Vegas)
1. Let $p = \textsc{Random}(10^{100})$
2. If $\textsc{isprime}(p)$=1 then $\textsc{Output}=p$ else goto 1

Key generation

Problem. Produce a random prime $p \approx 10^{100}$

Probabilistic algorithm (type Las Vegas)
1. Let $p = \textsc{Random}(10^{100})$
2. If ISPRIME(p)=1 then OUTPUT=p else goto 1

subproblems:

Key generation

Problem. Produce a random prime $p \approx 10^{100}$

Probabilistic algorithm (type Las Vegas)
1. Let $p = \textsc{Random}(10^{100})$
2. If isprime(p)=1 then Output=p else goto 1

subproblems:

A. How many iterations are necessary?

(i.e. how are primes distributes?)

Key generation

Problem. Produce a random prime $p \approx 10^{100}$

Probabilistic algorithm (type Las Vegas)
1. Let $p = \textsc{Random}(10^{100})$
2. If ISPRIME(p)=1 then OUTPUT=p else goto 1

subproblems:

A. How many iterations are necessary?

(i.e. how are primes distributes?)

B. How does one check if p is prime?

(i.e. how does one compute ISPRIME(p)?) $\rightsquigarrow$ Primality test

Key generation

Problem. Produce a random prime $p \approx 10^{100}$

Probabilistic algorithm (type Las Vegas)	
1.	Let $p = \textsc{Random}(10^{100})$
2.	If ISPRIME(p)=1 then OUTPUT=p else goto 1

subproblems:

A. How many iterations are necessary?

(i.e. how are primes distributes?)

B. How does one check if p is prime?

(i.e. how does one compute ISPRIME(p)?) $\rightsquigarrow$ Primality test

False Metropolitan Legend: Check primality is equivalent to factoring

A. Distribution of prime numbers

A. Distribution of prime numbers

$$\pi(x) = \#\{p \leq x \text{ t. c. } p \text{ is prime}\}$$

A. Distribution of prime numbers

$$\pi(x) = \#\{p \leq x \text{ t. c. } p \text{ is prime}\}$$

Theorem. (Hadamard - de la vallee Pussen - 1897)

$$\pi(x) \sim \frac{x}{\log x}$$

A. Distribution of prime numbers

$$\pi(x) = \#\{p \leq x \text{ t. c. } p \text{ is prime}\}$$

Theorem. (Hadamard - de la vallee Pussen - 1897)

$$\pi(x) \sim \frac{x}{\log x}$$

Quantitative version:

Theorem. (Rosser - Schoenfeld) if $x \geq 67$

$$\frac{x}{\log x - 1/2} < \pi(x) < \frac{x}{\log x - 3/2}$$

A. Distribution of prime numbers

$$\pi(x) = \#\{p \leq x \text{ t. c. } p \text{ is prime}\}$$

Theorem. (Hadamard - de la vallee Pussen - 1897)

$$\pi(x) \sim \frac{x}{\log x}$$

Quantitative version:

Theorem. (Rosser - Schoenfeld) if $x \geq 67$

$$\frac{x}{\log x - 1/2} < \pi(x) < \frac{x}{\log x - 3/2}$$

Therefore

$$0.0043523959267 < Prob\left((\textsc{Random}(10^{100}) = \texttt{prime}\right) < 0.004371422086$$

If P_k is the probability that among k random numbers $\leq 10^{100}$ there is a prime one, then

If P_k is the probability that among k random numbers $\leq 10^{100}$ there is a prime one, then

$$P_k = 1 - \left(1 - \frac{\pi(10^{100})}{10^{100}}\right)^k$$

If P_k is the probability that among k random numbers $\leq 10^{100}$ there is a prime one, then

$$P_k = 1 - \left(1 - \frac{\pi(10^{100})}{10^{100}}\right)^k$$

Therefore

$$0.663942 < P_{250} < 0.66554440$$

If P_k is the probability that among k random numbers$\leq 10^{100}$ there is a prime one, then

$$P_k = 1 - \left(1 - \frac{\pi(10^{100})}{10^{100}}\right)^k$$

Therefore

$$0.663942 < P_{250} < 0.66554440$$

To speed up the process: One can consider only odd random numbers not divisible by 3 nor by 5.

If P_k is the probability that among k random numbers$\leq 10^{100}$ there is a prime one, then

$$P_k = 1 - \left(1 - \frac{\pi(10^{100})}{10^{100}}\right)^k$$

Therefore

$$0.663942 < P_{250} < 0.66554440$$

To speed up the process: One can consider only odd random numbers not divisible by 3 nor by 5.

Let

$$\Psi(x, 30) = \#\{n \leq x \text{ s.t.} \gcd(n, 30) = 1\}$$

To speed up the process: One can consider only odd random numbers not divisible by 3 nor by 5.

To speed up the process: One can consider only odd random numbers not divisible by 3 nor by 5.

Let

$$\Psi(x,30) = \#\left\{n \le x \text{ s.t. } \gcd(n,30) = 1\right\}$$

then

To speed up the process: One can consider only odd random numbers not divisible by 3 nor by 5.

Let

$$\Psi(x, 30) = \#\{n \leq x \text{ s.t. } \gcd(n, 30) = 1\}$$

then

$$\frac{4}{15}x - 4 < \Psi(x, 30) < \frac{4}{15}x + 4$$

To speed up the process: One can consider only odd random numbers not divisible by 3 nor by 5.

Let

$$\Psi(x, 30) = \#\{n \leq x \text{ s.t. } \gcd(n, 30) = 1\}$$

then

$$\frac{4}{15}x - 4 < \Psi(x, 30) < \frac{4}{15}x + 4$$

Hence, if P'_k is the probability that among k random numbers $\leq 10^{100}$ coprime with 30, there is a prime one, then

To speed up the process: One can consider only odd random numbers not divisible by 3 nor by 5.

Let

$$\Psi(x,30) = \#\{n \leq x \text{ s.t.} \gcd(n,30)=1\}$$

then

$$\frac{4}{15}x - 4 < \Psi(x,30) < \frac{4}{15}x + 4$$

Hence, if P'_k is the probability that among k random numbers $\leq 10^{100}$ coprime with 30, there is a prime one, then

$$P'_k = 1 - \left(1 - \frac{\pi(10^{100})}{\Psi(10^{100},30)}\right)^k$$

To speed up the process: One can consider only odd random numbers not divisible by 3 nor by 5.

Let

$$\Psi(x,30) = \#\{n \leq x \text{ s.t. } \gcd(n,30)=1\}$$

then

$$\frac{4}{15}x - 4 < \Psi(x,30) < \frac{4}{15}x + 4$$

Hence, if P'_k is the probability that among k random numbers $\leq 10^{100}$ coprime with 30, there is a prime one, then

$$P'_k = 1 - \left(1 - \frac{\pi(10^{100})}{\Psi(10^{100},30)}\right)^k$$

and

$$0.98365832 < P'_{250} < 0.98395199$$

B. Primality test

B. Primality test

Fermat Little Theorem. If p is prime, $p \nmid a \in \mathbb{N}$

$$a^{p-1} \equiv 1 \bmod p$$

B. Primality test

Fermat Little Theorem. If p is prime, $p \nmid a \in \mathbb{N}$

$$a^{p-1} \equiv 1 \bmod p$$

NON-primality test

$$M \in \mathbb{Z}, \quad 2^{M-1} \not\equiv 1 \bmod M \Longrightarrow \quad M \text{composite!}$$

B. Primality test

Fermat Little Theorem. If p is prime, $p \nmid a \in \mathbb{N}$

$$a^{p-1} \equiv 1 \bmod p$$

NON-primality test

$$M \in \mathbb{Z}, \quad 2^{M-1} \not\equiv 1 \bmod M \implies \quad M \text{composite!}$$

EXAMPLE: $2^{RSA_{2048}-1} \not\equiv 1 \bmod RSA_{2048}$

Therefore RSA_{2048} is composite!

B. Primality test

Fermat Little Theorem. If p is prime, $p \nmid a \in \mathbb{N}$

$$a^{p-1} \equiv 1 \bmod p$$

NON-primality test

$$M \in \mathbb{Z}, \quad 2^{M-1} \not\equiv 1 \bmod M \implies \quad M \text{composite!}$$

EXAMPLE: $2^{RSA_{2048}-1} \not\equiv 1 \bmod RSA_{2048}$

Therefore RSA_{2048} is composite!

Fermat little Theorem does not invert. Infact

B. Primality test

Fermat Little Theorem. If p is prime, $p \nmid a \in \mathbb{N}$

$$a^{p-1} \equiv 1 \bmod p$$

NON-primality test

$$M \in \mathbb{Z}, \quad 2^{M-1} \not\equiv 1 \bmod M \Longrightarrow \quad M \text{composite!}$$

EXAMPLE: $2^{RSA_{2048}-1} \not\equiv 1 \bmod RSA_{2048}$

Therefore RSA_{2048} is composite!

Fermat little Theorem does not invert. Infact

$$2^{93960} \equiv 1 \pmod{93961} \quad \text{but} \quad 93961 = 7 \times 31 \times 433$$

Strong pseudo primes

Strong pseudo primes

From now on $m \equiv 3 \bmod 4$ (just to simplify the notation)

Strong pseudo primes

From now on $m \equiv 3 \bmod 4$ (just to simplify the notation)

Definition. $m \in \mathbb{N}$, $m \equiv 3 \bmod 4$, composite is said strong pseudo prime (SPSP) in base a if

$$a^{(m-1)/2} \equiv \pm 1 \pmod{m}.$$

Strong pseudo primes

From now on $m \equiv 3 \bmod 4$ (just to simplify the notation)

Definition. $m \in \mathbb{N}$, $m \equiv 3 \bmod 4$, composite is said strong pseudo prime (SPSP) in base a if

$$a^{(m-1)/2} \equiv \pm 1 \pmod{m}.$$

Note. If $p > 2$ prime $\Longrightarrow \quad a^{(p-1)/2} \equiv \pm 1 \pmod{p}$

Let $\boxed{\mathcal{S} = \{a \in \mathbb{Z}/m\mathbb{Z} \text{ s.t. } \gcd(m, a) = 1, a^{(m-1)/2} \equiv \pm 1 \pmod{m}\}}$

Strong pseudo primes

From now on $m \equiv 3 \bmod 4$ (just to simplify the notation)

Definition. $m \in \mathbb{N}$, $m \equiv 3 \bmod 4$, composite is said strong pseudo prime (SPSP) in base a if

$$a^{(m-1)/2} \equiv \pm 1 \pmod{m}.$$

Note. If $p > 2$ prime $\Longrightarrow$ $a^{(p-1)/2} \equiv \pm 1 \pmod{p}$

Let $\boxed{\mathcal{S} = \{a \in \mathbb{Z}/m\mathbb{Z} \text{ s.t. } \gcd(m, a) = 1, a^{(m-1)/2} \equiv \pm 1 \pmod{m}\}}$

①

②

③

④

Strong pseudo primes

From now on $m \equiv 3 \bmod 4$ (just to simplify the notation)

Definition. $m \in \mathbb{N}$, $m \equiv 3 \bmod 4$, composite is said strong pseudo prime (SPSP) in base a if

$$a^{(m-1)/2} \equiv \pm 1 \pmod{m}.$$

Note. If $p > 2$ prime $\Longrightarrow$ $a^{(p-1)/2} \equiv \pm 1 \pmod{p}$

Let $\boxed{\mathcal{S} = \{a \in \mathbb{Z}/m\mathbb{Z} \text{ s.t. } \gcd(m,a) = 1, a^{(m-1)/2} \equiv \pm 1 \pmod{m}\}}$

① $\mathcal{S} \subseteq (\mathbb{Z}/m\mathbb{Z})^*$ subgroup

②

③

④

Strong pseudo primes

From now on $m \equiv 3 \bmod 4$ (just to simplify the notation)

Definition. $m \in \mathbb{N}$, $m \equiv 3 \bmod 4$, composite is said strong pseudo prime (SPSP) in base a if

$$a^{(m-1)/2} \equiv \pm 1 \pmod{m}.$$

Note. If $p > 2$ prime $\Longrightarrow$ $a^{(p-1)/2} \equiv \pm 1 \pmod{p}$

Let $\boxed{\mathcal{S} = \{a \in \mathbb{Z}/m\mathbb{Z} \text{ s.t. } \gcd(m, a) = 1, a^{(m-1)/2} \equiv \pm 1 \pmod{m}\}}$

① $\mathcal{S} \subseteq (\mathbb{Z}/m\mathbb{Z})^*$ subgroup

② If m is composite $\Longrightarrow$ proper subgroup

③

④

Strong pseudo primes

From now on $m \equiv 3 \bmod 4$ (just to simplify the notation)

Definition. $m \in \mathbb{N}$, $m \equiv 3 \bmod 4$, composite is said strong pseudo prime (SPSP) in base a if

$$a^{(m-1)/2} \equiv \pm 1 \pmod{m}.$$

Note. If $p > 2$ prime $\Longrightarrow$ $a^{(p-1)/2} \equiv \pm 1 \pmod{p}$

Let $\boxed{\mathcal{S} = \{a \in \mathbb{Z}/m\mathbb{Z} \text{ s.t. } \gcd(m,a) = 1, a^{(m-1)/2} \equiv \pm 1 \pmod{m}\}}$

① $\mathcal{S} \subseteq (\mathbb{Z}/m\mathbb{Z})^*$ subgroup

② If m is composite $\Longrightarrow$ proper subgroup

③ If m is composite $\Longrightarrow$ $\#\mathcal{S} \leq \frac{\varphi(m)}{4}$

④

Strong pseudo primes

From now on $m \equiv 3 \bmod 4$ (just to simplify the notation)

Definition. $m \in \mathbb{N}$, $m \equiv 3 \bmod 4$, composite is said strong pseudo prime (SPSP) in base a if

$$a^{(m-1)/2} \equiv \pm 1 \pmod{m}.$$

Note. If $p > 2$ prime $\Longrightarrow \quad a^{(p-1)/2} \equiv \pm 1 \pmod{p}$

Let $\boxed{\mathcal{S} = \{a \in \mathbb{Z}/m\mathbb{Z} \text{ s.t. } \gcd(m,a) = 1, a^{(m-1)/2} \equiv \pm 1 \pmod{m}\}}$

① $\mathcal{S} \subseteq (\mathbb{Z}/m\mathbb{Z})^*$ subgroup

② If m is composite $\Longrightarrow$ proper subgroup

③ If m is composite $\Longrightarrow \quad \#\mathcal{S} \leq \frac{\varphi(m)}{4}$

④ If m is composite $\Longrightarrow \quad Prob(m \text{ PSPF in base } a) \leq 0,25$

Miller–Rabin primality test

Miller–Rabin primality test

Let $m \equiv 3 \bmod 4$

Miller–Rabin primality test

Let $m \equiv 3 \bmod 4$

Miller Rabin algorithm with k iterations
$N = (m-1)/2$
`for` $j = 0$ `to` k `do` $a =$`Random`(m)
`if` $a^N \not\equiv \pm 1 \bmod m$ `then OUPUT=(`m `composite): END`
`endfor OUTPUT=(`m `prime)`

Miller–Rabin primality test

Let $m \equiv 3 \bmod 4$

MILLER RABIN ALGORITHM WITH k ITERATIONS
$N = (m-1)/2$
`for` $j = 0$ `to` k `do` $a =$`Random`(m)
`if` $a^N \not\equiv \pm 1 \bmod m$ `then OUPUT=(`m `composite): END`
`endfor OUTPUT=(`m `prime)`

Monte Carlo primality test

Miller–Rabin primality test

Let $m \equiv 3 \bmod 4$

MILLER RABIN ALGORITHM WITH k ITERATIONS
$N = (m-1)/2$
`for` $j = 0$ `to` k `do` $a =$`Random`(m)
`if` $a^N \not\equiv \pm 1 \bmod m$ `then OUPUT=(`m `composite): END`
`endfor OUTPUT=(`m `prime)`

Monte Carlo primality test

$Prob$(Miller Rabin says m `prime` and m is composite) $\lessapprox \frac{1}{4^k}$

Miller–Rabin primality test

Let $m \equiv 3 \bmod 4$

Miller Rabin algorithm with k iterations
$N = (m-1)/2$
`for` $j = 0$ `to` k `do` $a =$`Random`(m)
`if` $a^N \not\equiv \pm 1 \bmod m$ `then OUPUT=(`m `composite): END`
`endfor OUTPUT=(`m `prime)`

Monte Carlo primality test

$Prob$(Miller Rabin says m `prime` and m is composite) $\lessapprox \frac{1}{4^k}$

In the real world, software uses Miller Rabin with $k = 10$

Deterministic primality tests

Deterministic primality tests

Theorem. (Miller, Bach) If m is composite, then

$$\textbf{GRH} \Rightarrow \ \exists a \leq 2\log^2 m \text{ s.t. } \ a^{(m-1)/2} \not\equiv \pm 1 \pmod{m}.$$

(i.e. *m is not SPSP in base a.*)

Deterministic primality tests

Theorem. (Miller, Bach) If m is composite, then

$$\textbf{GRH} \Rightarrow \ \exists a \leq 2\log^2 m \text{ s.t. } \ a^{(m-1)/2} \not\equiv \pm 1 \pmod{m}.$$

(i.e. *m is not SPSP in base a.*)

Consequence: "Miller–Rabin de–randomizes on GRH" ($m \equiv 3 \bmod 4$)

Deterministic primality tests

Theorem. (Miller, Bach) If m is composite, then

$$\textbf{GRH} \Rightarrow \ \exists a \leq 2\log^2 m \text{ s.t. } \ a^{(m-1)/2} \not\equiv \pm 1 \pmod{m}.$$

(i.e. *m is not SPSP in base a.*)

Consequence: "Miller–Rabin de–randomizes on GRH" ($m \equiv 3 \bmod 4$)

`for`	$a = 2$ `to` $2\log^2 m$	`do`	
	`if` $a^{(m-1)/2} \not\equiv \pm 1 \bmod m$	`then`	
		`OUPUT=(`m `composite):`	`END`
`endfor`		`OUTPUT=(`m `prime)`	

Deterministic primality tests

Theorem. (Miller, Bach) If m is composite, then

$$\textbf{GRH} \Longrightarrow \exists a \leq 2\log^2 m \text{ s.t. } a^{(m-1)/2} \not\equiv \pm 1 \pmod{m}.$$

(i.e. *m is not SPSP in base a.*)

Consequence: "Miller–Rabin de–randomizes on GRH" ($m \equiv 3 \bmod 4$)

`for`	$a = 2$ `to` $2\log^2 m$	`do`	
	`if` $a^{(m-1)/2} \not\equiv \pm 1 \bmod m$	`then`	
		`OUPUT=(`m `composite):`	`END`
`endfor`		`OUTPUT=(`m `prime)`	

Deterministic Polynomial time algorithm

Deterministic primality tests

Theorem. (Miller, Bach) If m is composite, then

$$\textbf{GRH} \Longrightarrow \exists a \leq 2\log^2 m \text{ s.t. } a^{(m-1)/2} \not\equiv \pm 1 \pmod{m}.$$

(i.e. *m is not SPSP in base a.*)

Consequence: "Miller–Rabin de–randomizes on GRH" ($m \equiv 3 \bmod 4$)

```
for     a = 2 to 2 log^2 m                 do
        if a^((m-1)/2) ≢ ±1 mod m          then
                                           OUPUT=(m composite):  END
endfor                                     OUTPUT=(m prime)
```

Deterministic Polynomial time algorithm

It runs in $O(\log^5 m)$ operations in $\mathbb{Z}/m\mathbb{Z}$.

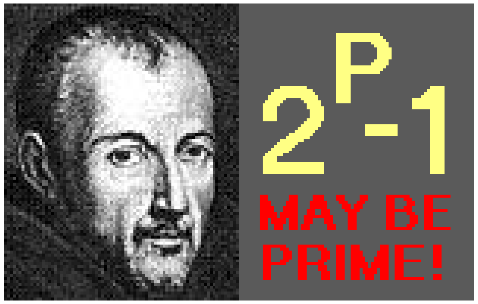
Certified prime records

Certified prime records

- $2^{57885161} - 1$, 17425170 digits (discovered in 01/2014)
- $2^{43112609} - 1$, 12978189 digits (discovered in 2008)
- $2^{42643801} - 1$, 12837064 digits (discovered in 2009)
- $2^{37156667} - 1$, 11185272 digits (discovered in 2008)
- $2^{32582657} - 1$, 9808358 digits (discovered in 2006)
- $2^{30402457} - 1$, 9152052 digits (discovered in 2005)
- $2^{25964951} - 1$, 7816230 digits (discovered in 2005)
- $2^{24036583} - 1$, 6320430 digits (discovered in 2004)
- $2^{20996011} - 1$, 6320430 digits (discovered in 2003)
- $2^{13466917} - 1$, 4053946 digits (discovered in 2001)
- $2^{6972593} - 1$, 2098960 digits (discovered in 1999)
- $5359 \times 2^{5054502} + 1$, 1521561 digits (discovered in 2003)

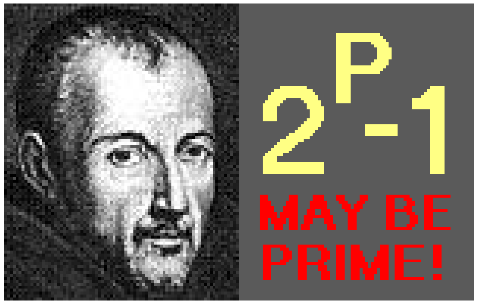
Great Internet Mersenne Prime Search (GIMPS)

Great Internet Mersenne Prime Search (GIMPS)



The Great Internet Mersenne Prime Search (GIMPS) is a collaborative project of volunteers who use freely available software to search for Mersenne prime numbers (i.e. prime numbers of the form $2^p - 1$ (p prime)).

Great Internet Mersenne Prime Search (GIMPS)



The Great Internet Mersenne Prime Search (GIMPS) is a collaborative project of volunteers who use freely available software to search for Mersenne prime numbers (i.e. prime numbers of the form $2^p - 1$ (p prime)).

The project was founded by George Woltman in January 1996.

The AKS deterministic primality test

The AKS deterministic primality test

Department of Computer Science & Engineering,
I.I.T. Kanpur, Agost 8, 2002.

The AKS deterministic primality test

Department of Computer Science & Engineering,
I.I.T. Kanpur, Agost 8, 2002.

Nitin Saxena, Neeraj Kayal and Manindra Agarwal

The AKS deterministic primality test

Department of Computer Science & Engineering,
I.I.T. Kanpur, Agost 8, 2002.

Nitin Saxena, Neeraj Kayal and Manindra Agarwal
New deterministic, polynomial–time, primality test.

The AKS deterministic primality test

Department of Computer Science & Engineering,
I.I.T. Kanpur, Agost 8, 2002.

Nitin Saxena, Neeraj Kayal and Manindra Agarwal
New deterministic, polynomial–time, primality test.

Solves #1 open question in computational number theory

The AKS deterministic primality test

Department of Computer Science & Engineering,
I.I.T. Kanpur, Agost 8, 2002.

Nitin Saxena, Neeraj Kayal and Manindra Agarwal

New deterministic, polynomial–time, primality test.

Solves #1 open question in computational number theory

http://www.cse.iitk.ac.in/news/primality.html

How does the AKS work?

How does the AKS work?

Theorem. (AKS) Let $n \in \mathbb{N}$. Assume q, r primes, $S \subseteq \mathbb{N}$ finite:

- $q | r - 1$;
- $n^{(r-1)/q} \bmod r \notin \{0, 1\}$;
- $\gcd(n, b - b') = 1, \quad \forall b, b' \in S$ (distinct);
- $\binom{q+\#S-1}{\#S} \geq n^{2\lfloor\sqrt{r}\rfloor}$;
- $(x + b)^n = x^n + b$ in $\mathbb{Z}/n\mathbb{Z}[x]/(x^r - 1), \quad \forall b \in S$;

Then n is a power of a prime Bernstein formulation

How does the AKS work?

Theorem. (AKS) Let $n \in \mathbb{N}$. Assume q, r primes, $S \subseteq \mathbb{N}$ finite:

- $q | r - 1$;
- $n^{(r-1)/q} \bmod r \notin \{0, 1\}$;
- $\gcd(n, b - b') = 1, \quad \forall b, b' \in S$ (distinct);
- $\binom{q+\#S-1}{\#S} \geq n^{2\lfloor\sqrt{r}\rfloor}$;
- $(x + b)^n = x^n + b$ in $\mathbb{Z}/n\mathbb{Z}[x]/(x^r - 1), \quad \forall b \in S$;

Then n is a power of a prime Bernstein formulation

Fouvry Theorem (1985) $\Longrightarrow \quad \exists r \approx \log^6 n, s \approx \log^4 n$

How does the AKS work?

Theorem. (AKS) Let $n \in \mathbb{N}$. Assume q, r primes, $S \subseteq \mathbb{N}$ finite:

- $q | r - 1$;
- $n^{(r-1)/q} \bmod r \notin \{0, 1\}$;
- $\gcd(n, b - b') = 1, \quad \forall b, b' \in S$ (distinct);
- $\binom{q+\#S-1}{\#S} \geq n^{2\lfloor\sqrt{r}\rfloor}$;
- $(x + b)^n = x^n + b$ in $\mathbb{Z}/n\mathbb{Z}[x]/(x^r - 1), \quad \forall b \in S$;

Then n is a power of a prime   Bernstein formulation

Fouvry Theorem (1985) $\Longrightarrow \quad \exists r \approx \log^6 n, s \approx \log^4 n$

$\Longrightarrow$ AKS runs in $O(\log^{15} n)$ operations in $\mathbb{Z}/n\mathbb{Z}$.

How does the AKS work?

Theorem. (AKS) Let $n \in \mathbb{N}$. Assume q, r primes, $S \subseteq \mathbb{N}$ finite:

- $q | r - 1$;
- $n^{(r-1)/q} \bmod r \notin \{0, 1\}$;
- $\gcd(n, b - b') = 1, \quad \forall b, b' \in S$ (distinct);
- $\binom{q+\#S-1}{\#S} \geq n^{2\lfloor\sqrt{r}\rfloor}$;
- $(x + b)^n = x^n + b$ in $\mathbb{Z}/n\mathbb{Z}[x]/(x^r - 1), \quad \forall b \in S$;

Then n is a power of a prime — Bernstein formulation

Fouvry Theorem (1985) $\Longrightarrow$ $\exists r \approx \log^6 n, s \approx \log^4 n$

$\Longrightarrow$ AKS runs in $O(\log^{15} n)$ operations in $\mathbb{Z}/n\mathbb{Z}$.

Many simplifications and improvements: Bernstein, Lenstra, Pomerance.....

Why is RSA safe?

Why is RSA safe?

☞

☞

☞

Why is RSA safe?

☞ It is clear that if Charles can factor M,

☞

☞

Why is RSA safe?

- It is clear that if Charles can factor M,
  then he can also compute $\varphi(M)$ and then also d so to decrypt messages
-
-

Why is RSA safe?

- ☞ It is clear that if Charles can factor M,
  then he can also compute $\varphi(M)$ and then also d so to decrypt messages
- ☞ Computing $\varphi(M)$ is equivalent to completely factor M. In fact
- ☞

Why is RSA safe?

- ☞ It is clear that if Charles can factor M, then he can also compute $\varphi(M)$ and then also d so to decrypt messages
- ☞ Computing $\varphi(M)$ is equivalent to completely factor M. In fact
$$p, q = \frac{M - \varphi(M) + 1 \pm \sqrt{(M - \varphi(M) + 1)^2 - 4M}}{2}$$
- ☞

Why is RSA safe?

- ☞ It is clear that if Charles can factor M,
  then he can also compute $\varphi(M)$ and then also d so to decrypt messages
- ☞ Computing $\varphi(M)$ is equivalent to completely factor M. In fact

$$p, q = \frac{M - \varphi(M) + 1 \pm \sqrt{(M - \varphi(M) + 1)^2 - 4M}}{2}$$

- ☞ **RSA Hypothesis.** The only way to compute efficiently

Why is RSA safe?

- It is clear that if Charles can factor M, then he can also compute $\varphi(M)$ and then also d so to decrypt messages
- Computing $\varphi(M)$ is equivalent to completely factor M. In fact

$$p, q = \frac{M - \varphi(M) + 1 \pm \sqrt{(M - \varphi(M) + 1)^2 - 4M}}{2}$$

- **RSA Hypothesis.** The only way to compute efficiently

$$x^{1/e} \bmod M, \quad \forall x \in \mathbb{Z}/M\mathbb{Z}$$

Why is RSA safe?

- It is clear that if Charles can factor M,
  then he can also compute $\varphi(M)$ and then also d so to decrypt messages
- Computing $\varphi(M)$ is equivalent to completely factor M. In fact
$$p, q = \frac{M - \varphi(M) + 1 \pm \sqrt{(M - \varphi(M) + 1)^2 - 4M}}{2}$$
- **RSA Hypothesis.** The only way to compute efficiently
$$x^{1/e} \bmod M, \quad \forall x \in \mathbb{Z}/M\mathbb{Z}$$
  (i.e. decrypt messages) is to factor M

Why is RSA safe?

☞ It is clear that if Charles can factor M,
then he can also compute $\varphi(M)$ and then also d so to decrypt messages

☞ Computing $\varphi(M)$ is equivalent to completely factor M. In fact

$$p, q = \frac{M - \varphi(M) + 1 \pm \sqrt{(M - \varphi(M) + 1)^2 - 4M}}{2}$$

☞ **RSA Hypothesis.** The only way to compute efficiently

$$x^{1/e} \bmod M, \quad \forall x \in \mathbb{Z}/M\mathbb{Z}$$

(i.e. decrypt messages) is to factor M

In other words

Why is RSA safe?

- It is clear that if Charles can factor M,
  then he can also compute $\varphi(M)$ and then also d so to decrypt messages
- Computing $\varphi(M)$ is equivalent to completely factor M. In fact

$$p, q = \frac{M - \varphi(M) + 1 \pm \sqrt{(M - \varphi(M) + 1)^2 - 4M}}{2}$$

- **RSA Hypothesis.** The only way to compute efficiently

$$x^{1/e} \bmod M, \quad \forall x \in \mathbb{Z}/M\mathbb{Z}$$

(i.e. decrypt messages) is to factor M

In other words

The two problems are polynomially equivalent

Two kinds of Cryptography

Two kinds of Cryptography

- **Private key (or symmetric)**
  - Lucifer
  - DES
  - AES

Two kinds of Cryptography

- **Private key (or symmetric)**
  - Lucifer
  - DES
  - AES
- **Public key**
  - RSA
  - Diffie–Hellmann
  - Knapsack
  - NTRU

Another quotation!!!

Have you ever noticed that there's no attempt being made to find really large numbers that aren't prime. I mean, wouldn't you like to see a news report that says "Today the Department of Computer Sciences at the University of Washington annouced that $2^{58,111,625,031}+8$ *is even". This is the largest non-prime yet reported.*

- University of Washington (Bathroom Graffiti)