COMP170
Discrete Mathematical Tools for Computer Science
The RSA Algorithm

Version 2.1 Last updated, September 5, 2008

Discrete Math for Computer Science
K. Bogart, C. Stein and R.L. Drysdale
Sections 2.3, 2.4, pp. 72-86

2.3 The RSA Cryptosystem

2.3 The RSA Cryptosystem

- Assorted Tools and Definitions

2.3 The RSA Cryptosystem

- Assorted Tools and Definitions
- Exponentiation mod n

2.3 The RSA Cryptosystem

- Assorted Tools and Definitions
- Exponentiation mod n
- The Rules of Exponents

2.3 The RSA Cryptosystem

- Assorted Tools and Definitions
- Exponentiation mod n
- The Rules of Exponents
- Fermat's Little Theorem

2.3 The RSA Cryptosystem

- Assorted Tools and Definitions
- Exponentiation mod n
- The Rules of Exponents
- Fermat's Little Theorem
- The RSA Cryptosystem

2.3 The RSA Cryptosystem

- Assorted Tools and Definitions
- Exponentiation mod n
- The Rules of Exponents
- Fermat's Little Theorem
- The RSA Cryptosystem
- Practical Aspects of Exponentiation mod n

2.3 The RSA Cryptosystem

- Assorted Tools and Definitions
- Exponentiation mod n
- The Rules of Exponents
- Fermat's Little Theorem
- The RSA Cryptosystem
- Practical Aspects of Exponentiation mod n
- The Chinese Remainder Theorem

Consider multiplication in Z_7

$\cdot_7$	1	2	3	4	5	6
1	1	2	3	4	5	6
2	2	4	6	1	3	5
3	3	6	2	5	1	4
4	4	1	5	2	6	3
5	5	3	1	6	4	2
6	6	5	4	3	2	1

Consider multiplication in Z_7

For every nonzero $a \in Z_7$, the function $f_a(x) = x \cdot_7 a$ is one-to-one and therefore a permutation of $Z_7 - \{0\}$, i.e., every row is a permutation.

$\cdot_7$	1	2	3	4	5	6
1	1	2	3	4	5	6
2	2	4	6	1	3	5
3	3	6	2	5	1	4
4	4	1	5	2	6	3
5	5	3	1	6	4	2
6	6	5	4	3	2	1

Consider multiplication in Z_7

For every nonzero $a \in Z_7$, the function $f_a(x) = x \cdot_7 a$ is one-to-one and therefore a permutation of $Z_7 - \{0\}$, i.e., every row is a permutation.

$\cdot_7$	1	2	3	4	5	6
1	1	2	3	4	5	6
2	2	4	6	1	3	5
3	3	6	2	5	1	4
4	4	1	5	2	6	3
5	5	3	1	6	4	2
6	6	5	4	3	2	1

Lemma 2.20: Let p be a prime number. For any nonzero number $a \in Z_p$, the function $f_a(x) = x \cdot_p a$ is 1-to-1. In particular, the numbers, $1 \cdot_p a,\ 2 \cdot_p a,\ \ldots,\ (p-1) \cdot_p a$, are a permutation of the set $\{1, 2, \ldots, p-1\}$.

Lemma 2.20: Let p be a prime number. For any nonzero number $a \in Z_p$, the function $f_a(x) = x \cdot_p a$ is 1-to-1. In particular, the numbers, $1 \cdot_p a$, $2 \cdot_p a$, $\ldots$, $(p-1) \cdot_p a$, are a permutation of the set $\{1, 2, \ldots, p-1\}$.

Proof: Suppose $f_a(x)$ is not 1-to-1. Then there are $x \neq y$ with $f_a(x) = f_a(y)$. Since p is prime, Corollary 2.17 tells us that there is $a^{-1} \in Z_p$ s.t. $a \cdot_p a^{-1} = 1$.

Multiplying the two sides by a^{-1}gives

$$\begin{aligned} x = (x \cdot_p a) \cdot_p a^{-1} &= f_a(x) \cdot_p a^{-1} \\ &= f_a(y) \cdot_p a^{-1} = (y \cdot_p a) \cdot_p a^{-1} = y \end{aligned}$$

Contradiction!

Lemma 2.20: Let p be a prime number. For any nonzero number $a \in Z_p$, the function $f_a(x) = x \cdot_p a$ is 1-to-1. In particular, the numbers, $1 \cdot_p a,\ 2 \cdot_p a,\ \ldots,\ (p-1) \cdot_p a$, are a permutation of the set $\{1, 2, \ldots, p-1\}$.

Proof: Suppose $f_a(x)$ is not 1-to-1. Then there are $x \neq y$ with $f_a(x) = f_a(y)$. Since p is prime, Corollary 2.17 tells us that there is $a^{-1} \in Z_p$ s.t. $a \cdot_p a^{-1} = 1$.

Multiplying the two sides by a^{-1}gives

$$\begin{aligned} x = (x \cdot_p a) \cdot_p a^{-1} &= f_a(x) \cdot_p a^{-1} \\ &= f_a(y) \cdot_p a^{-1} = (y \cdot_p a) \cdot_p a^{-1} = y \end{aligned}$$

Contradiction! $\Rightarrow$ Then $f_a(x)$ is 1-to-1

- A one-to-one function $f : X \to Y$
  is a one-way function
  if knowing $f(x)$ does not provide you with enough information to *efficiently* recover x.

- A one-to-one function $f : X \to Y$
  is a one-way function
  if knowing $f(x)$ does not provide you with enough information to *efficiently* recover x.

- Note that the definition of one-way function has been intentionally left quite imprecise. If f is one-to-one, then the inverse g of f with $g(f(x)) = x$ always exists.
  Knowing that g *exists*, though, does not always help in calculating $g(u)$. For a given u, $g(u)$ might be hard to calculate.

- A one-to-one function $f : X \to Y$
  is a one-way function
  if knowing $f(x)$ does not provide you with enough information to *efficiently* recover x.

- Note that the definition of one-way function has been intentionally left quite imprecise. If f is one-to-one, then the
  inverse g of f with $g(f(x)) = x$ always exists.

  Knowing that g *exists*, though, does not always help in calculating $g(u)$. For a given u, $g(u)$ might be hard to calculate.

- For public-key cryptography,
  the public encoding function, P_B, needs to be one-way.
  The secret decoding function, S_B, is actually
  an efficient way of calculating the inverse of P_B.
  This efficient way is only available to the
  "owner" who constructed P_B.

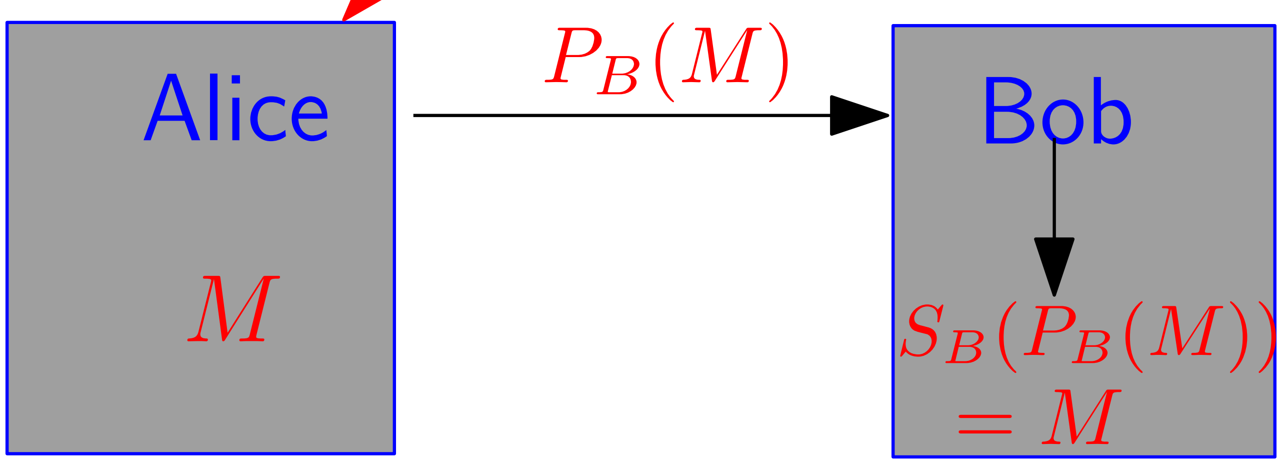
Recall the Public-Key Setup

i) Alice wants to send M to Bob

ii) In public directory, Alice looks up Bob's Public Key, P_B

iii) Alice sends $P_B(M)$ to Bob

iv) Bob uses his Secret Key, S_B to decrypt $M = S_B(P_B(M))$



2.3 The RSA Cryptosystem

- Assorted Tools and Definitions
- Exponentiation mod n
- The Rules of Exponents
- Fermat's Little Theorem
- The RSA Cryptosystem
- Practical Aspects of Exponentiation mod n
- The Chinese Remainder Theorem

Exponentiation mod n

Exponentiation mod n

Last time, we considered encryption using modular addition and multiplication, and have seen weaknesses of both.

Exponentiation mod n

Last time, we considered encryption using modular addition and multiplication, and have seen weaknesses of both.

We now consider using *exponentiation* for encryption.

Exponentiation mod n

Last time, we considered encryption using modular addition and multiplication, and have seen weaknesses of both.

We now consider using *exponentiation* for encryption.

Exponentiation in Z_n is the main idea behind RSA encryption:

Exponentiation mod n

Last time, we considered encryption using modular addition and multiplication, and have seen weaknesses of both.

We now consider using *exponentiation* for encryption.

Exponentiation in Z_n is the main idea behind RSA encryption:

By Lemma 2.3, if $a \in Z_n$, then

$$a^j \bmod n \;=\; \underbrace{a \cdot_n a \cdot_n \cdots \cdot_n a}_{j \text{ factors}}.$$

Exponentiation mod n

Last time, we considered encryption using modular addition and multiplication, and have seen weaknesses of both.

We now consider using *exponentiation* for encryption.

Exponentiation in Z_n is the main idea behind RSA encryption:

By Lemma 2.3, if $a \in Z_n$, then

$$a^j \text{ mod } n \;=\; \underbrace{a \cdot_n a \cdot_n \cdots \cdot_n a}_{j \text{ factors}}.$$

a^j mod n is the product in Z_n of j factors, each equal to a.

2.3 The RSA Crytosystem

- Assorted Tools and Definitions
- Exponentiation mod n
- The Rules of Exponents
- Fermat's Little Theorem
- The RSA Cryptosystem
- Practical Aspects of Exponentiation mod n
- The Chinese Remainder Theorem

The Rules of Exponentiation

The Rules of Exponentiation

From Lemma 2.3 and exponentiation for integers, we have

Lemma 2.19:
For any $a \in Z_n$ and any nonnegative integers i, j,

a) $$(a^i \bmod n) \cdot_n (a^j \bmod n) = a^{i+j} \bmod n$$

b) $$(a^i \bmod n)^j \bmod n = a^{ij} \bmod n$$

The Rules of Exponentiation

From Lemma 2.3 and exponentiation for integers, we have

Lemma 2.19:
For any $a \in Z_n$ and any nonnegative integers i, j,

a) $$(a^i \bmod n) \cdot_n (a^j \bmod n) = a^{i+j} \bmod n$$

b) $$(a^i \bmod n)^j \bmod n = a^{ij} \bmod n$$

Examples:

$$\begin{array}{lll} 3^2 & = & 9 \\ 3^4 & = & 81 \\ 3^6 & = & 729 \\ 3^8 & = & 6561 \end{array} \qquad \begin{array}{lll} 3^2 \bmod 7 & = & 2 \\ 3^4 \bmod 7 & = & 4 \\ 3^6 \bmod 7 & = & 1 \\ 3^8 \bmod 7 & = & 2 \end{array}$$

The Rules of Exponentiation

From Lemma 2.3 and exponentiation for integers, we have

Lemma 2.19:
For any $a \in Z_n$ and any nonnegative integers i, j,

a) $$(a^i \bmod n) \cdot_n (a^j \bmod n) = a^{i+j} \bmod n$$

b) $$(a^i \bmod n)^j \bmod n = a^{ij} \bmod n$$

Examples:

$$\begin{aligned} 3^2 &= 9 & 3^2 \bmod 7 &= 2 \\ 3^4 &= 81 & 3^4 \bmod 7 &= 4 \\ 3^6 &= 729 & 3^6 \bmod 7 &= 1 \\ 3^8 &= 6561 & 3^8 \bmod 7 &= 2 \end{aligned}$$

a) $$1 = \left(3^2 \bmod 7\right) \cdot_7 \left(3^4 \bmod 7\right) = 3^6 \bmod 7$$

The Rules of Exponentiation

From Lemma 2.3 and exponentiation for integers, we have

Lemma 2.19:
For any $a \in Z_n$ and any nonnegative integers i, j,

a) $$(a^i \bmod n) \cdot_n (a^j \bmod n) = a^{i+j} \bmod n$$

b) $$(a^i \bmod n)^j \bmod n = a^{ij} \bmod n$$

Examples:

$$\begin{array}{lcl} 3^2 & = & 9 \\ 3^4 & = & 81 \\ 3^6 & = & 729 \\ 3^8 & = & 6561 \end{array} \qquad \begin{array}{lcl} 3^2 \bmod 7 & = & 2 \\ 3^4 \bmod 7 & = & 4 \\ 3^6 \bmod 7 & = & 1 \\ 3^8 \bmod 7 & = & 2 \end{array}$$

a) $$1 = \left(3^2 \bmod 7\right) \cdot_7 \left(3^4 \bmod 7\right) = 3^6 \bmod 7$$

b) $$2 = 16 \bmod 7 = \left(3^4 \bmod 7\right)^2 \bmod 7 = 3^8 \bmod 7$$

2.3 The RSA Cryptosystem

- Assorted Tools and Definitions
- Exponentiation mod n
- The Rules of Exponents
- Fermat's Little Theorem
- The RSA Cryptosystem
- Practical Aspects of Exponentiation mod n
- The Chinese Remainder Theorem

Choose $x \in Z_7$. Now examine the sequence $x^0, x^1, x^2, x^3, \ldots$. Do you see a pattern?

$\cdot_7$	1	2	3	4	5	6
1	1	2	3	4	5	6
2	2	4	6	1	3	5
3	3	6	2	5	1	4
4	4	1	5	2	6	3
5	5	3	1	6	4	2
6	6	5	4	3	2	1

Choose $x \in Z_7$. Now examine the sequence $x^0, x^1, x^2, x^3, \ldots$. Do you see a pattern?

$\cdot_7$	1	2	3	4	5	6
1	1	2	3	4	5	6
2	2	4	6	1	3	5
3	3	6	2	5	1	4
4	4	1	5	2	6	3
5	5	3	1	6	4	2
6	6	5	4	3	2	1

x	x^0	x^1	x^2	x^3	x^4	x^5	x^6
1	1	1	1	1	1	1	1
2	1	2	4	1	2	4	1
3	1	3	2	6	4	5	1
4	1	4	2	1	4	2	1
5	1	5	4	6	2	3	1
6	1	6	1	6	1	6	1

$\cdot_7$	1	2	3	4	5	6
1	1	2	3	4	5	6
2	2	4	6	1	3	5
3	3	6	2	5	1	4
4	4	1	5	2	6	3
5	5	3	1	6	4	2
6	6	5	4	3	2	1

Choose $x \in Z_7$. Now examine the sequence $x^0, x^1, x^2, x^3, \ldots$. Do you see a pattern?

x	x^0	x^1	x^2	x^3	x^4	x^5	x^6
1	1	1	1	1	1	1	1
2	1	2	4	1	2	4	1
3	1	3	2	6	4	5	1
4	1	4	2	1	4	2	1
5	1	5	4	6	2	3	1
6	1	6	1	6	1	6	1

For *every* $x \in Z_7$, the sequence starts *cycling*.
In particular, for every $x \in Z_7$, we have
$x^0 = 1 = x^6 = x^{7-1}$.

Theorem 2.21 (Fermat's Little Theorem):
Let p be a prime number. Then, for every nonzero $a \in Z_p$,
$$a^{p-1} \bmod p = 1.$$

Theorem 2.21 (Fermat's Little Theorem):
Let p be a prime number. Then, for every nonzero $a \in Z_p$,

$$a^{p-1} \bmod p = 1.$$

Proof: Since p is prime, Lemma 2.20 tells us that

$\boxed{1 \cdot_p a,\ 2 \cdot_p a,\ \ldots,\ (p-1) \cdot_p a}$ are a permutation of $\boxed{1, 2, \ldots, p-1}$.

Theorem 2.21 (Fermat's Little Theorem):
Let p be a prime number. Then, for every nonzero $a \in Z_p$,

$$a^{p-1} \bmod p = 1.$$

Proof: Since p is prime, Lemma 2.20 tells us that

$\boxed{1 \cdot_p a,\ 2 \cdot_p a,\ \ldots,\ (p-1) \cdot_p a}$ are a permutation of $\boxed{1, 2, \ldots, p-1}$.

$$\Rightarrow 1 \cdot_p 2 \cdot_p \cdots \cdot_p (p-1) = (1 \cdot_p a) \cdot_p (2 \cdot_p a) \cdot_p \cdots \cdot_p ((p-1) \cdot_p a)$$

Theorem 2.21 (Fermat's Little Theorem):
Let p be a prime number. Then, for every nonzero $a \in Z_p$,

$$a^{p-1} \bmod p = 1.$$

Proof: Since p is prime, Lemma 2.20 tells us that

$\boxed{1 \cdot_p a,\ 2 \cdot_p a,\ \ldots,\ (p-1) \cdot_p a}$ are a permutation of $\boxed{1, 2, \ldots, p-1}$.

$$\begin{aligned} \Rightarrow 1 \cdot_p 2 \cdot_p \cdots \cdot_p (p-1) &= (1 \cdot_p a) \cdot_p (2 \cdot_p a) \cdot_p \cdots \cdot_p ((p-1) \cdot_p a) \\ &= [1 \cdot_p 2 \cdot_p \cdots \cdot_p (p-1)] \cdot_p \left(a^{p-1} \bmod p\right) \end{aligned}$$

Theorem 2.21 (Fermat's Little Theorem):
Let p be a prime number. Then, for every nonzero $a \in Z_p$,

$$a^{p-1} \bmod p = 1.$$

Proof: Since p is prime, Lemma 2.20 tells us that

$\boxed{1 \cdot_p a,\ 2 \cdot_p a,\ \ldots,\ (p-1) \cdot_p a}$ are a permutation of $\boxed{1, 2, \ldots, p-1}$.

$$\Rightarrow 1 \cdot_p 2 \cdot_p \cdots \cdot_p (p-1) = (1 \cdot_p a) \cdot_p (2 \cdot_p a) \cdot_p \cdots \cdot_p ((p-1) \cdot_p a)$$
$$= [1 \cdot_p 2 \cdot_p \cdots \cdot_p (p-1)] \cdot_p \left(a^{p-1} \bmod p\right)$$

Now let $x = 1 \cdot_p 2 \cdot_p \cdots \cdot_p (p-1)$.

The equation above is $x = x \cdot_p \left(a^{p-1} \bmod p\right)$

Since p is prime, x^{-1} exists in Z_p. So,

Theorem 2.21 (Fermat's Little Theorem):
Let p be a prime number. Then, for every nonzero $a \in Z_p$,

$$a^{p-1} \bmod p = 1.$$

Proof: Since p is prime, Lemma 2.20 tells us that

$\boxed{1 \cdot_p a,\ 2 \cdot_p a,\ \ldots,\ (p-1) \cdot_p a}$ are a permutation of $\boxed{1, 2, \ldots, p-1}$.

$$\Rightarrow 1 \cdot_p 2 \cdot_p \cdots \cdot_p (p-1) = (1 \cdot_p a) \cdot_p (2 \cdot_p a) \cdot_p \cdots \cdot_p ((p-1) \cdot_p a)$$
$$= [1 \cdot_p 2 \cdot_p \cdots \cdot_p (p-1)] \cdot_p \left(a^{p-1} \bmod p\right)$$

Now let $x = 1 \cdot_p 2 \cdot_p \cdots \cdot_p (p-1)$.

The equation above is $x = x \cdot_p \left(a^{p-1} \bmod p\right)$

Since p is prime, x^{-1} exists in Z_p. So,

$$1 = x^{-1} \cdot_p x = x^{-1} \cdot_p x \cdot_p \left(a^{p-1} \bmod p\right)$$
$$= a^{p-1} \bmod p$$

Theorem 2.21 (Fermat's Little Theorem):
Let p be a prime number. Then, for every nonzero $a \in Z_p$,

$$a^{p-1} \bmod p = 1.$$

Theorem 2.21 (Fermat's Little Theorem):
Let p be a prime number. Then, for every nonzero $a \in Z_p$,

$$a^{p-1} \bmod p = 1.$$

This implies

Corollary 2.22 (Fermat's Little Theorem, Version 2)
Let p be a prime number. Then, for every positive integer a that is not a multiple of p,

$$a^{p-1} \bmod p = 1.$$

Theorem 2.21 (Fermat's Little Theorem):
Let p be a prime number. Then, for every nonzero $a \in Z_p$,

$$a^{p-1} \bmod p = 1.$$

This implies

Corollary 2.22 (Fermat's Little Theorem, Version 2)
Let p be a prime number. Then, for every positive integer a that is not a multiple of p,

$$a^{p-1} \bmod p = 1.$$

Proof:
Direct application of Lemma 2.3, because if we replace a with $a \bmod p$, then Theorem 2.21 applies.

Theorem 2.21 (Fermat's Little Theorem):
Let p be a prime number. Then, for every nonzero $a \in Z_p$,

$$a^{p-1} \bmod p = 1.$$

Theorem 2.21 (Fermat's Little Theorem):
Let p be a prime number. Then, for every nonzero $a \in Z_p$,

$$a^{p-1} \bmod p = 1.$$

This also implies

Corollary 2.X1 Let p be a prime number. Let m be a non-negative integer. Then, for every positive integer a that is not a multiple of p,

$$a^m \bmod p = a^{(m \bmod (p-1))} \bmod p.$$

Theorem 2.21 (Fermat's Little Theorem):
Let p be a prime number. Then, for every nonzero $a \in Z_p$,

$$a^{p-1} \bmod p = 1.$$

This also implies

Corollary 2.X1 Let p be a prime number. Let m be a non-negative integer. Then, for every positive integer a that is not a multiple of p,

$$a^m \bmod p = a^{(m \bmod (p-1))} \bmod p.$$

Example: $a = 5,\ p = 7, m = 15$

$$\Rightarrow \qquad a^{15} \bmod 7 = a^{(2\cdot 6+3)} \bmod 7 = a^3 \bmod 7 = 6$$

Pierre de Fermat

French Mathematician

b. 1601. d. 1665

Pierre de Fermat

French Mathematician

b. 1601. d. 1665

Worked in probability theory and number theory (which he helped found)

Pierre de Fermat

French Mathematician

b. 1601. d. 1665

Worked in probability theory and number theory (which he helped found)

Most famous for Fermat's Last Theorem. This says that the equation $x^n + y^n = z^n$ has no solution for integers x, y, z, n with $n > 2$. Fermat had written in the margin of one of his math books that

I have discovered a truly remarkable proof which this margin is too small to contain.

Pierre de Fermat

French Mathematician

b. 1601. d. 1665

Worked in probability theory and number theory (which he helped found)

Most famous for Fermat's Last Theorem. This says that the equation $x^n + y^n = z^n$ has no solution for integers x, y, z, n with $n > 2$. Fermat had written in the margin of one of his math books that

> *I have discovered a truly remarkable proof which this margin is too small to contain.*

It took mathematicians more than 300 years to "rediscover" a proof for this (if you believe that Fermat ever had one). Andrew Wiles finally managed to prove this in 1994.

2.3 The RSA Cryptosystem

- Assorted Tools and Definitions
- Exponentiation mod n
- The Rules of Exponents
- Fermat's Little Theorem
- The RSA Cryptosystem
- Practical Aspects of Exponentiation mod n
- The Chinese Remainder Theorem

RSA are the initials of three Computer Scientists, Ron Rivest, Adi Shamir and Len Adleman, who discovered their algorithm when they were working together at MIT in 1977.

It is now known that Cifford Cocks, a mathematician working for Government Communications Headquarters (GCHQ), the secret coding agency in Britan, independently discovered this earlier, in 1973, but did not publish his work. This fact was not known until certain secret British documents were declassified in 1997.

The RSA Cryptosystem

The RSA Cryptosystem Finally!!

The RSA Cryptosystem Finally!!

Bob's RSA Key Choice Algorithm

The RSA Cryptosystem Finally!!

Bob's RSA Key Choice Algorithm

(1) Choose 2 large prime numbers p and q

The RSA Cryptosystem Finally!!

Bob's RSA Key Choice Algorithm

(1) Choose 2 large prime numbers p and q > 150 digits

The RSA Cryptosystem Finally!!

Bob's RSA Key Choice Algorithm

(1) Choose 2 large prime numbers p and q > 150 digits

(2) Set $n = pq$ and $T = (p-1)(q-1)$

The RSA Cryptosystem Finally!!

Bob's RSA Key Choice Algorithm

(1) Choose 2 large prime numbers p and q > 150 digits

(2) Set $n = pq$ and $T = (p-1)(q-1)$

(3) Choose $e \neq 1$ so that $gcd(e, T) = 1$

The RSA Cryptosystem Finally!!

Bob's RSA Key Choice Algorithm

(1) Choose 2 large prime numbers p and q — > 150 digits

(2) Set $n = pq$ and $T = (p-1)(q-1)$

(3) Choose $e \neq 1$ so that $gcd(e, T) = 1$ — Any prime that doesn't divide T

The RSA Cryptosystem Finally!!

Bob's RSA Key Choice Algorithm

(1) Choose 2 large prime numbers p and q — > 150 digits

(2) Set $n = pq$ and $T = (p-1)(q-1)$

(3) Choose $e \neq 1$ so that $gcd(e, T) = 1$ — Any prime that doesn't divide T

(4) Calculate $d = e^{-1} \bmod T$

The RSA Cryptosystem Finally!!

Bob's RSA Key Choice Algorithm

(1) Choose 2 large prime numbers p and q — > 150 digits

(2) Set $n = pq$ and $T = (p-1)(q-1)$

(3) Choose $e \neq 1$ so that $gcd(e, T) = 1$ — Any prime that doesn't divide T

(4) Calculate $d = e^{-1} \bmod T$ — Extended GCD Alg

The RSA Cryptosystem Finally!!

Bob's RSA Key Choice Algorithm

(1) Choose 2 large prime numbers p and q — > 150 digits

(2) Set $n = pq$ and $T = (p-1)(q-1)$

(3) Choose $e \neq 1$ so that $gcd(e, T) = 1$ — Any prime that doesn't divide T

(4) Calculate $d = e^{-1} \bmod T$ — Extended GCD Alg

(5) Publish e, n as public key

The RSA Cryptosystem Finally!!

Bob's RSA Key Choice Algorithm

(1) Choose 2 large prime numbers p and q > 150 digits

(2) Set $n = pq$ and $T = (p-1)(q-1)$

(3) Choose $e \neq 1$ so that $gcd(e, T) = 1$ Any prime that doesn't divide T

(4) Calculate $d = e^{-1} \bmod T$ Extended GCD Alg

(5) Publish e, n as public key

(6) Keep d as secret key

Alice-send-message-to-Bob(x) $(0 \leq x < n)$

(1) Alice does:
(2) Read public directory for Bob's public keys e and n.
(3) Compute $y = x^e \bmod n$
(4) Send y to Bob

(5) Bob does:
(6) Receive y from Alice
(7) Compute $z = y^d \bmod n$, using secret key d
(8) Read z

Alice-send-message-to-Bob(x) $(0 \le x < n)$

(1) Alice does:

(2) Read public directory for Bob's public keys e and n.

(3)   Compute $y = x^e \bmod n$

(4)   Send y to Bob

(5) Bob does:

(6) Receive y from Alice

(7) Compute $z = y^d \bmod n$, using secret key d

(8) Read z

To show that the RSA cryptosystem works — that is, that it allows us to correcty decode encoded messages — we must show that $x = z$, i.e., for all x, $0 \le x < n$,

$$x = (x^e \bmod n)^d \bmod n = x^{ed} \bmod n$$

Story so far: We have (*)

Want to prove that,

if $0 \leq x < n$

$$x = x^{ed} \bmod n$$

(*) $\boxed{\begin{array}{l} p, q \text{ prime} \\ n = pq \\ T = (p-1)(q-1) \\ e \text{ s.t. } gcd(e, T) = 1 \\ d = e^{-1} \bmod T \end{array}}$

Story so far: We have (*)

Want to prove that,

if $0 \le x < n$

$$x \;=\; x^{ed} \bmod n$$

(*) $\boxed{\begin{array}{l} p, q \text{ prime} \\ n = pq \\ T = (p-1)(q-1) \\ e \text{ s.t. } gcd(e,T) = 1 \\ d = e^{-1} \bmod T \end{array}}$

Plan

(1) Proving that $x \bmod p = x^{ed} \bmod p$ for all x

(2) Proving that $x \bmod q = x^{ed} \bmod q$ for all x

(3) Showing that, if $0 \le x < n$, (1) + (2) imply

$$x = x^{ed} \pmod{n}$$

Story so far: We have (*)

Want to prove

(1) $x \bmod p = \ x^{ed} \bmod p$

(*) $\boxed{\begin{array}{l} p, q \text{ prime} \\ n = pq \\ T = (p-1)(q-1) \\ e \text{ s.t. } gcd(e, T) = 1 \\ d = e^{-1} \bmod T \end{array}}$

Story so far: We have (*)
Want to prove

(1) $x \bmod p = \; x^{ed} \bmod p$

(*) $\boxed{\begin{array}{l} p, q \text{ prime} \\ n = pq \\ T = (p-1)(q-1) \\ e \text{ s.t. } gcd(e,T) = 1 \\ d = e^{-1} \bmod T \end{array}}$

$ed \bmod T = 1$ so there is some k such that $ed = 1 + kT$ and

$$\begin{aligned} x^{ed} \bmod p &= x^{1+k(q-1)(p-1)} \bmod p \\ &= x\left(x^{k(q-1)}\right)^{p-1} \bmod p \end{aligned}$$

There are two possible cases

(a) $x^{k(q-1)}$ is a multiple of p

(b) $x^{k(q-1)}$ is not a multiple of p

$$
\begin{aligned}
x^{ed} \bmod p &= x^{1+k(q-1)(p-1)} \bmod p \\
&= x\left(x^{k(q-1)}\right)^{p-1} \bmod p
\end{aligned}
$$

There are two possible cases

(a) $x^{k(q-1)}$ is a multiple of p

(b) $x^{k(q-1)}$ is not a multiple of p

$$
\begin{aligned}
x^{ed} \bmod p &= x^{1+k(q-1)(p-1)} \bmod p \\
&= x\left(x^{k(q-1)}\right)^{p-1} \bmod p
\end{aligned}
$$

There are two possible cases

(a) $x^{k(q-1)}$ is a multiple of p

(b) $x^{k(q-1)}$ is not a multiple of p

(a) If $x^{k(q-1)}$ is a multiple of p

$\Rightarrow$ since p is prime, x is also a multiple of p.

$\Rightarrow$ $x^{ed} \bmod p = 0 = x \bmod p$

$$\begin{aligned} x^{ed} \bmod p &= x^{1+k(q-1)(p-1)} \bmod p \\ &= x\left(x^{k(q-1)}\right)^{p-1} \bmod p \end{aligned}$$

There are two possible cases

(a) $x^{k(q-1)}$ is a multiple of p

(b) $x^{k(q-1)}$ is not a multiple of p

(a) If $x^{k(q-1)}$ is a multiple of p

$\Rightarrow$ since p is prime, x is also a multiple of p.

$\Rightarrow$ $\boxed{x^{ed} \bmod p = 0 = x \bmod p}$

$$
\begin{aligned}
x^{ed} \bmod p &= x^{1+k(q-1)(p-1)} \bmod p \\
&= x\left(x^{k(q-1)}\right)^{p-1} \bmod p
\end{aligned}
$$

There are two possible cases

$\checkmark$(a) $x^{k(q-1)}$ is a multiple of p $\Rightarrow x^{ed} \bmod p = 0 = x \bmod p$

(b) $x^{k(q-1)}$ is not a multiple of p

$$\begin{aligned} x^{ed} \bmod p &= x^{1+k(q-1)(p-1)} \bmod p \\ &= x\left(x^{k(q-1)}\right)^{p-1} \bmod p \end{aligned}$$

There are two possible cases

√(a) $x^{k(q-1)}$ is a multiple of p $\Rightarrow x^{ed} \bmod p = 0 = x \bmod p$

(b) $x^{k(q-1)}$ is not a multiple of p

(b) If $a = x^{k(q-1)}$ is not a multiple of p then

$$x^{ed} \bmod p = x\left(x^{k(q-1)}\right)^{p-1} \bmod p$$

$$
\begin{aligned}
x^{ed} \bmod p &= x^{1+k(q-1)(p-1)} \bmod p \\
&= x\left(x^{k(q-1)}\right)^{p-1} \bmod p
\end{aligned}
$$

There are two possible cases

✓(a) $x^{k(q-1)}$ is a multiple of p $\quad\Rightarrow x^{ed} \bmod p = 0 = x \bmod p$

(b) $x^{k(q-1)}$ is not a multiple of p

(b) If $a = x^{k(q-1)}$ is not a multiple of p then

$$x^{ed} \bmod p = x\left(x^{k(q-1)}\right)^{p-1} \bmod p$$

Fermat's little thm:
If $p \not| a$ then
$a^{p-1} \bmod p = 1$

$$\begin{aligned} x^{ed} \bmod p &= x^{1+k(q-1)(p-1)} \bmod p \\ &= x\left(x^{k(q-1)}\right)^{p-1} \bmod p \end{aligned}$$

There are two possible cases

✓(a) $x^{k(q-1)}$ is a multiple of p $\Rightarrow x^{ed} \bmod p = 0 = x \bmod p$

(b) $x^{k(q-1)}$ is not a multiple of p

(b) If $a = x^{k(q-1)}$ is not a multiple of p then

$$\begin{aligned} x^{ed} \bmod p = x\left(x^{k(q-1)}\right)^{p-1} \bmod p \\ = x \cdot 1 \bmod p \end{aligned}$$

Fermat's little thm:
If $p \not| a$ then
$a^{p-1} \bmod p = 1$

$$
\begin{aligned}
x^{ed} \bmod p &= x^{1+k(q-1)(p-1)} \bmod p \\
&= x\left(x^{k(q-1)}\right)^{p-1} \bmod p
\end{aligned}
$$

There are two possible cases

✓(a) $x^{k(q-1)}$ is a multiple of p $\Rightarrow x^{ed} \bmod p = 0 = x \bmod p$

(b) $x^{k(q-1)}$ is not a multiple of p

(b) If $a = x^{k(q-1)}$ is not a multiple of p then

$$
\begin{aligned}
x^{ed} \bmod p &= x\left(x^{k(q-1)}\right)^{p-1} \bmod p \\
&= x \cdot 1 \bmod p \\
&= x \bmod p
\end{aligned}
$$

Fermat's little thm:
If $p \nmid a$ then
$a^{p-1} \bmod p = 1$

$$
\begin{aligned}
x^{ed} \bmod p &= x^{1+k(q-1)(p-1)} \bmod p \\
&= x\left(x^{k(q-1)}\right)^{p-1} \bmod p
\end{aligned}
$$

There are two possible cases

√(a) $x^{k(q-1)}$ is a multiple of p $\Rightarrow x^{ed} \bmod p = 0 = x \bmod p$

√(b) $x^{k(q-1)}$ is not a multiple of p $\Rightarrow x^{ed} \bmod p = x \bmod p$

$$\begin{aligned} x^{ed} \bmod p &= x^{1+k(q-1)(p-1)} \bmod p \\ &= x\left(x^{k(q-1)}\right)^{p-1} \bmod p \end{aligned}$$

There are two possible cases

√(a) $x^{k(q-1)}$ is a multiple of p $\Rightarrow x^{ed} \bmod p = 0 = x \bmod p$

√(b) $x^{k(q-1)}$ is not a multiple of p $\Rightarrow x^{ed} \bmod p = x \bmod p$

We have therefore just finished proving that, for all x

$$x^{ed} \bmod p = x \bmod p$$

Story so far: We have (*)

Want to prove that,

if $0 \le x < n$

$$x = x^{ed} \bmod n$$

(*) $\boxed{\begin{array}{l} p, q \text{ prime} \\ n = pq \\ T = (p-1)(q-1) \\ e \text{ s.t. } gcd(e, T) = 1 \\ d = e^{-1} \bmod T \end{array}}$

Story so far: We have (*)

Want to prove that,

if $0 \leq x < n$

$$x \;=\; x^{ed} \bmod n$$

(*) $\boxed{\begin{array}{l} p, q \text{ prime} \\ n = pq \\ T = (p-1)(q-1) \\ e \text{ s.t. } gcd(e,T) = 1 \\ d = e^{-1} \bmod T \end{array}}$

Plan

Story so far: We have (*)

Want to prove that,

if $0 \le x < n$

$$x = x^{ed} \bmod n$$

(*)
$$\boxed{\begin{array}{l} p, q \text{ prime} \\ n = pq \\ T = (p-1)(q-1) \\ e \text{ s.t. } gcd(e,T) = 1 \\ d = e^{-1} \bmod T \end{array}}$$

Plan

√ (1) Proved that $x \bmod p = x^{ed} \bmod p$

Story so far: We have (*)

Want to prove that,

if $0 \leq x < n$

$$x = x^{ed} \bmod n$$

(*) $\boxed{\begin{array}{l} p, q \text{ prime} \\ n = pq \\ T = (p-1)(q-1) \\ e \text{ s.t. } gcd(e, T) = 1 \\ d = e^{-1} \bmod T \end{array}}$

Plan

√ (1) Proved that $x \bmod p = x^{ed} \bmod p$

(2) Proved that $x \bmod q = x^{ed} \bmod q$

Story so far: We have (*)

Want to prove that,

if $0 \leq x < n$

$$x = x^{ed} \bmod n$$

(*) $\boxed{\begin{array}{l} p, q \text{ prime} \\ n = pq \\ T = (p-1)(q-1) \\ e \text{ s.t. } gcd(e,T) = 1 \\ d = e^{-1} \bmod T \end{array}}$

Plan

√ (1) Proved that $x \bmod p = x^{ed} \bmod p$

√ (2) Proved that $x \bmod q = x^{ed} \bmod q$

Exact same proof as (1)

Story so far: We have (*)

Want to prove that,

if $0 \le x < n$

$$x = x^{ed} \bmod n$$

(*)
$$\boxed{\begin{array}{l} p, q \text{ prime} \\ n = pq \\ T = (p-1)(q-1) \\ e \text{ s.t. } gcd(e,T) = 1 \\ d = e^{-1} \bmod T \end{array}}$$

Plan

√ (1) Proved that $x \bmod p = x^{ed} \bmod p$

√ (2) Proved that $x \bmod q = x^{ed} \bmod q$

Exact same proof as (1)

(3) Need to show that (1) + (2) imply

$$x = x^{ed} \bmod n$$

Quick review of prime number properties

If p and q are both prime numbers and *both* divide z
then pq divides z

Example:

$p = 3$, $q = 11$, $z = 99$

$3, 11$ both divide 99 $\quad\Rightarrow\quad$ $33 = pq$ also divides 99

Quick review of prime number properties

If p and q are both prime numbers and *both* divide z
then pq divides z

Example:

$p = 3,\ q = 11,\ z = 99$

$3, 11$ both divide 99 $\quad\Rightarrow\quad$ $33 = pq$ also divides 99

Note that if p, q are *not* prime this is not necessarily true

Example:

$p = 6,\ q = 15,\ z = 60$

$6, 15$ both divide 60 but $90 = pq$ does not divide 60

Story so far: We have (*)

Want to prove that,

if $0 \leq x < n$

$$x = x^{ed} \bmod n$$

(*) $\boxed{\begin{array}{l} p, q \text{ prime} \\ n = pq \\ T = (p-1)(q-1) \\ e \text{ s.t. } gcd(e, T) = 1 \\ d = e^{-1} \bmod T \end{array}}$

Story so far: We have (*)

Want to prove that,

if $0 \leq x < n$

$$x = x^{ed} \bmod n$$

(*) $\boxed{\begin{array}{l} p, q \text{ prime} \\ n = pq \\ T = (p-1)(q-1) \\ e \text{ s.t. } gcd(e, T) = 1 \\ d = e^{-1} \bmod T \end{array}}$

Know that $x^{ed} \bmod p = x \bmod p$ and $x^{ed} \bmod q = x \bmod q$

Story so far: We have (*)

Want to prove that,

if $0 \leq x < n$

$$x \;=\; x^{ed} \bmod n$$

(*) $\boxed{\begin{array}{l} p, q \text{ prime} \\ n = pq \\ T = (p-1)(q-1) \\ e \text{ s.t. } gcd(e, T) = 1 \\ d = e^{-1} \bmod T \end{array}}$

Know that $x^{ed} \bmod p = x \bmod p$ and $x^{ed} \bmod q = x \bmod q$

Then $\begin{array}{l} v = x^{ed} - x = ip \\ v = x^{ed} - x = jq \end{array}$ for some integers i, j

Story so far: We have (*)

Want to prove that,

if $0 \le x < n$

$$x = x^{ed} \bmod n$$

(*)
$$\boxed{\begin{array}{l} p, q \text{ prime} \\ n = pq \\ T = (p-1)(q-1) \\ e \text{ s.t. } gcd(e, T) = 1 \\ d = e^{-1} \bmod T \end{array}}$$

Know that $x^{ed} \bmod p = x \bmod p$ and $x^{ed} \bmod q = x \bmod q$

Then
$$v = x^{ed} - x = ip$$
$$v = x^{ed} - x = jq$$
for some integers i, j

Then primes p, q *both* divide v, so $n = pq$ divides v

Story so far: We have (*)

Want to prove that,

if $0 \leq x < n$

$$x \;=\; x^{ed} \bmod n$$

(*)

p, q prime

$n = pq$

$T = (p-1)(q-1)$

e s.t. $gcd(e, T) = 1$

$d = e^{-1} \bmod T$

Know that $x^{ed} \bmod p = x \bmod p$ and $x^{ed} \bmod q = x \bmod q$

Then
$$v = x^{ed} - x = ip$$
$$v = x^{ed} - x = jq$$
for some integers i, j

Then primes p, q *both* divide v, so $n = pq$ divides v

Then $x^{ed} = kn + x$ for some k. Since $0 \leq x < n$,

$$x^{ed} \bmod n = x$$

We just saw that if

Alice-send-message-to-Bob(x) $(0 \leq x < n)$

(1) Alice does:
(2) Read public directory for Bob's public keys e and n.
(3) Compute $y = x^e \bmod n$
(4) Send y to Bob

(5) Bob does:
(6) Receive y from Alice
(7) Compute $z = y^d \bmod n$, using secret key d
(8) Read z

We just saw that if

Alice-send-message-to-Bob(x) $(0 \le x < n)$

(1) Alice does:

(2) Read public directory for Bob's public keys e and n.

(3) Compute $y = x^e \bmod n$

(4) Send y to Bob

(5) Bob does:

(6) Receive y from Alice

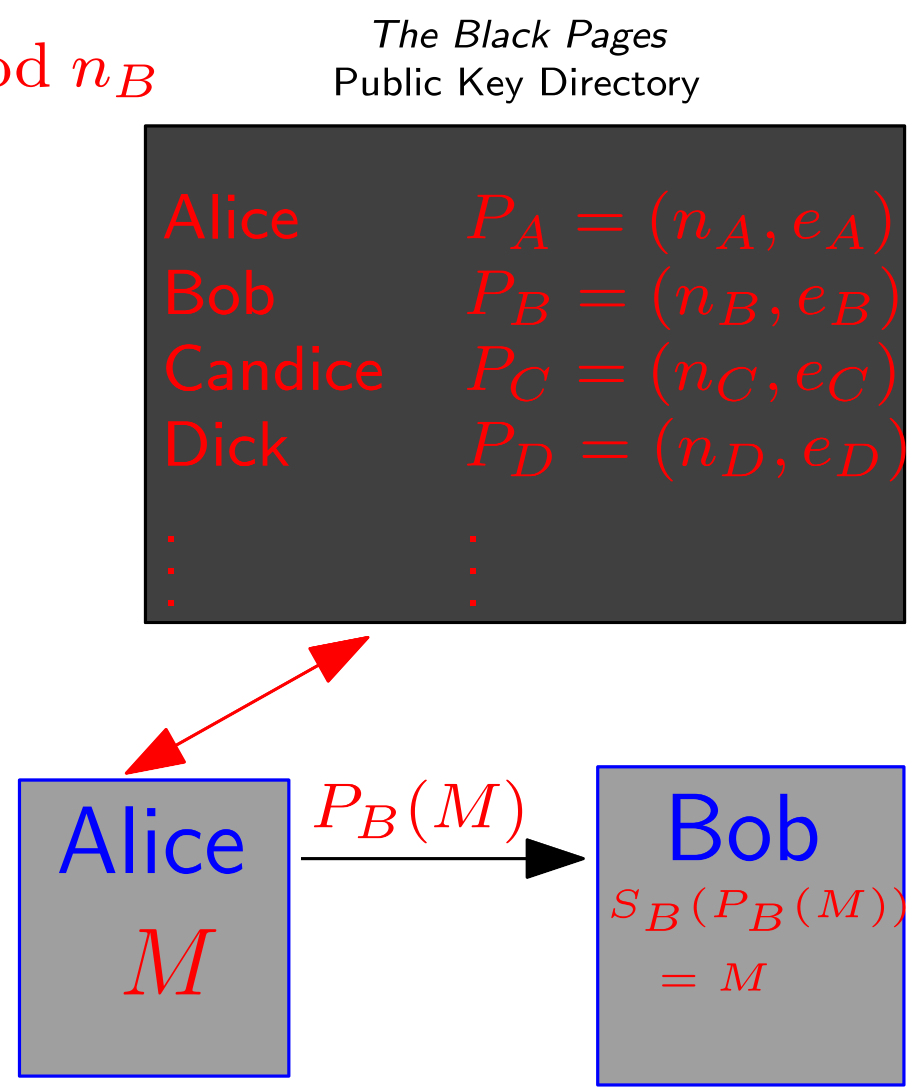
(7) Compute $z = y^d \bmod n$, using secret key d

(8) Read z

$$\Rightarrow \quad z = x^{ed} \bmod n = x$$

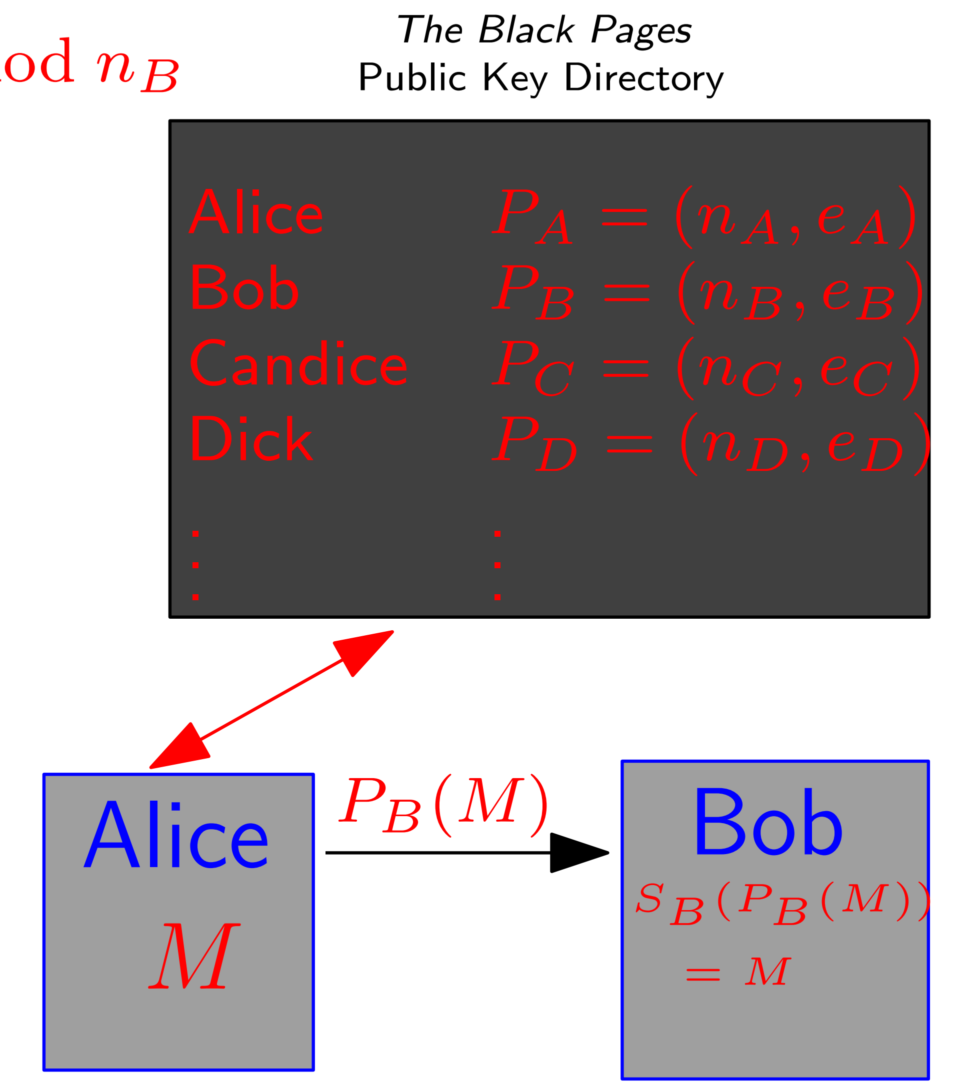
Theorem 2.23 (Rivest, Shamir, and Adleman)
The RSA procedure for encoding and decoding messages works correctly.

$P_B(M) = M^{e_B} \bmod n_B \quad S_B(Y) = Y^{d_B} \bmod n_B$



Theorem 2.23 (Rivest, Shamir, and Adleman) The RSA procedure for encoding and decoding messages works correctly.

$P_B(M) = M^{e_B} \bmod n_B \quad S_B(Y) = Y^{d_B} \bmod n_B$

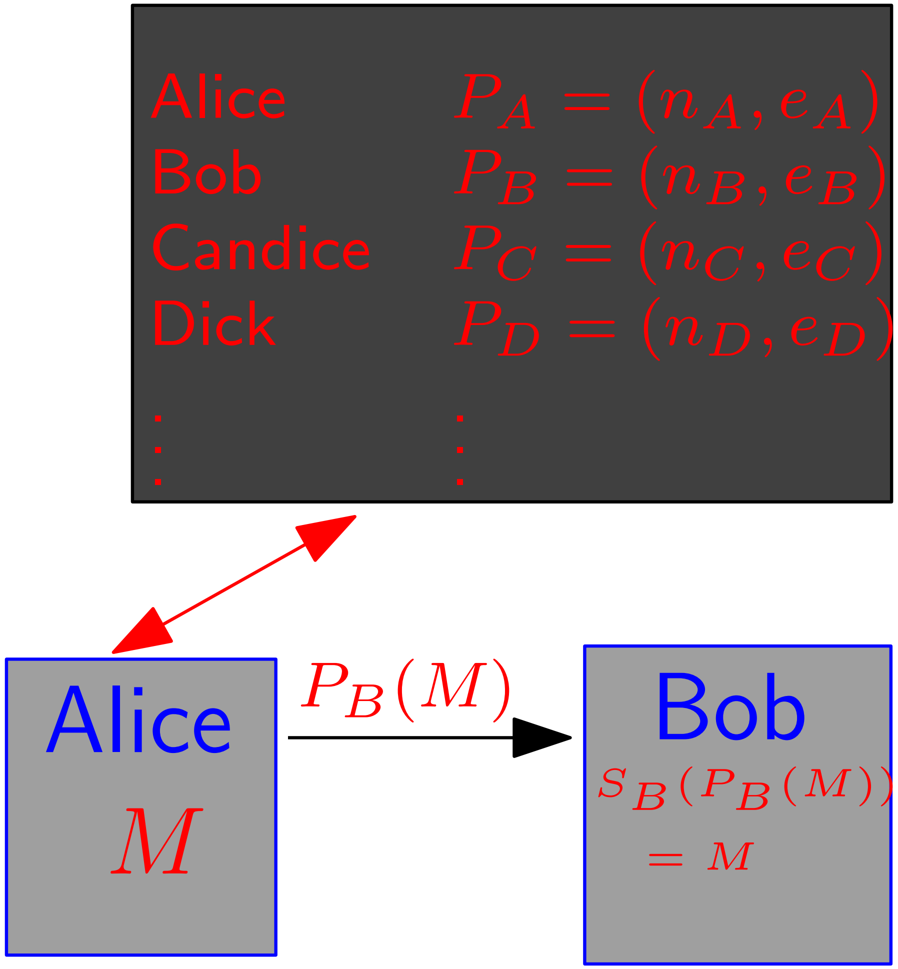


Why is this secret?

Theorem 2.23 (Rivest, Shamir, and Adleman) The RSA procedure for encoding and decoding messages works correctly.

$P_B(M) = M^{e_B} \bmod n_B \quad S_B(Y) = Y^{d_B} \bmod n_B$

The Black Pages
Public Key Directory



Why is this secret?

We claim that someone (adversary) who knows the public information n, e and $M^e \bmod n$ can not figure out M.

- To show that the RSA cryptosystem is secure, we must argue that an adversary (eavesdropper) who knows n, e, and $M^e \bmod n$, but does not know p, q or d, can not easily compute M.

- To show that the RSA cryptosystem is secure, we must argue that an adversary (eavesdropper) who knows n, e, and $M^e \bmod n$, but does not know p, q or d, can not easily compute M.

- At present, nobody knows a quick scheme for computing eth roots $\mathrm{mod} n$, for an arbitrary n. Thus, the adversary will not be able to work backwards and find M from $M^e \bmod n$. Thus, as far as we know, modular exponentiation is an example of a one-way function and the RSA system is secure.

- To show that the RSA cryptosystem is secure, we must argue that an adversary (eavesdropper) who knows n, e, and $M^e \bmod n$, but does not know p, q or d, can not easily compute M.

- At present, nobody knows a quick scheme for computing e^{th} roots $\text{mod} n$, for an arbitrary n. Thus, the adversary will not be able to work backwards and find M from $M^e \bmod n$. Thus, as far as we know, modular exponentiation is an example of a one-way function and the RSA system is secure.

- But, the adversary knows n and knows that n is the product of two prime numbers. Can't he just factor n to find p, q s.t. $n = pq$. Once he knows p, q he can construct d by himself and read the message!

- To show that the RSA cryptosystem is secure, we must argue that an adversary (eavesdropper) who knows n, e, and $M^e \bmod n$, but does not know p, q or d, can not easily compute M.

- At present, nobody knows a quick scheme for computing e^{th} roots $\bmod n$, for an arbitrary n. Thus, the adversary will not be able to work backwards and find M from $M^e \bmod n$. Thus, as far as we know, modular exponentiation is an example of a one-way function and the RSA system is secure.

- But, the adversary knows n and knows that n is the product of two prime numbers. Can't he just factor n to find p, q s.t. $n = pq$. Once he knows p, q he can construct d by himself and read the message!

  No!!. Nobody knows how to factor quickly!

- Think about this for a moment

- Think about this for a moment
- Most e-commerce and computer security is based on RSA or similar schemes

- Think about this for a moment
- Most e-commerce and computer security is based on RSA or similar schemes
- If you knew how to factor numbers into their prime components quickly, you could break RSA

- Think about this for a moment
- Most e-commerce and computer security is based on RSA or similar schemes
- If you knew how to factor numbers into their prime components quickly, you could break RSA
- So, if you could figure out a quick factoring scheme, you could break most modern computer security

- Think about this for a moment
- Most e-commerce and computer security is based on RSA or similar schemes
- If you knew how to factor numbers into their prime components quickly, you could break RSA
- So, if you could figure out a quick factoring scheme, you could break most modern computer security
- *Note: Although nobody knows how to factor quickly we don't have any proof that factoring* **must** *be slow. It's possible that there's a fast factorization algorithm out there that no one has found yet . . . .*

RSA Example

Parameters: $p = 5, q = 11$: $\Rightarrow$ $T = (p-1)*(q-1) = 40$.

RSA Example

Parameters: $p = 5, q = 11$: $\Rightarrow$ $T = (p-1)*(q-1) = 40$.

Let $e = 7$; using the extended GCD algorithm on $7, 40$ we find that $7 \cdot 23 - 40 \cdot 4 = 1$ so $d = 23$ is the multiplicative inverse of $e \bmod 40$:

RSA Example

Parameters: $p = 5, q = 11$: $\Rightarrow$ $T = (p-1)*(q-1) = 40$.

Let $e = 7$; using the extended GCD algorithm on $7, 40$ we find that $7 \cdot 23 - 40 \cdot 4 = 1$ so $d = 23$ is the multiplicative inverse of $e \bmod 40$:

Examples: For the given $n = p \cdot q = 55$, $e = 7$, $d = 23$

for $x = 12$:
$12^7 \bmod 55 = 35831808 \bmod 55 = 23$ and
$23^{23} \bmod 55 = 20880467999847912034355032910567 \bmod 55 = 12$.

RSA Example

Parameters: $p = 5, q = 11$: $\Rightarrow$ $T = (p-1)*(q-1) = 40$.

Let $e = 7$; using the extended GCD algorithm on $7, 40$ we find that $7 \cdot 23 - 40 \cdot 4 = 1$ so $d = 23$ is the multiplicative inverse of $e \bmod 40$:

Examples: For the given $n = p \cdot q = 55$, $e = 7$, $d = 23$

for $x = 12$:
$12^7 \bmod 55 = 35831808 \bmod 55 = 23$ and
$23^{23} \bmod 55 = 20880467999847912034355032910567 \bmod 55 = 12.$

for $x = 15$:
$15^7 \bmod 55 = 170859375 \bmod 55 = 5$ and
$5^{23} \bmod 55 = 11920928955078125 \bmod 55 = 15$

RSA Example

Parameters: $p = 5, q = 11$: $\Rightarrow$ $T = (p-1)*(q-1) = 40$.

Let $e = 7$; using the extended GCD algorithm on $7, 40$ we find that $7 \cdot 23 - 40 \cdot 4 = 1$ so $d = 23$ is the multiplicative inverse of $e \bmod 40$:

Examples: For the given $n = p \cdot q = 55$, $e = 7$, $d = 23$

for $x = 12$:
$12^7 \bmod 55 = 35831808 \bmod 55 = 23$ and
$23^{23} \bmod 55 = 20880467999847912034355032910567 \bmod 55 = 12.$

for $x = 15$:
$15^7 \bmod 55 = 170859375 \bmod 55 = 5$ and
$5^{23} \bmod 55 = 11920928955078125 \bmod 55 = 15$

for $x = 22$:
$22^7 \bmod 55 = 2494357888 \bmod 55 = 33$ and
$33^{23} \bmod 55 = 84298649517881922539738734663399137 \bmod 55 = 22$

2.3 The RSA Cryptosystem

- Assorted Tools and Definitions
- Exponentiation mod n
- The Rules of Exponents
- Fermat's Little Theorem
- The RSA Cryptosystem
- Practical Aspects of Exponentiation mod n
- The Chinese Remainder Theorem

Practical Aspects of Exponentiation $\bmod n$

Practical Aspects of Exponentiation $\bmod n$

Suppose you want to calculate $a^e \bmod n$

Sizes to right not unusual in RSA

a – 150 digits

e – 10^{120}, 121 digits

n – 150 digits

Practical Aspects of Exponentiation $\bmod n$

Suppose you want to calculate $a^e \bmod n$

Sizes to right not unusual in RSA

a – 150 digits

e – 10^{120}, 121 digits

n – 150 digits

- 1^{st} try: Calculate a^e (how) and then take $\bmod n$

Practical Aspects of Exponentiation $\bmod n$

Suppose you want to calculate $a^e \bmod n$

Sizes to right not unusual in RSA

a – 150 digits

e – 10^{120}, 121 digits

n – 150 digits

- 1^{st} try: Calculate a^e (how) and then take $\bmod n$

  No! a^e has $\sim 150 \cdot 10^{120}$ digits.

  It won't fit in our computer!

Practical Aspects of Exponentiation $\bmod n$

Suppose you want to calculate $a^e \bmod n$

Sizes to right not unusual in RSA

a – 150 digits

e – 10^{120}, 121 digits

n – 150 digits

- 1^{st} try: Calculate a^e (how) and then take $\bmod n$

  No! a^e has $\sim 150 \cdot 10^{120}$ digits.

  It won't fit in our computer!

- 2^{nd} try: Iteratively calculate values between 0 and n using

  $a^{i+1} \bmod n = a\left(a^i \bmod n\right) \bmod n$

Practical Aspects of Exponentiation $\bmod n$

Suppose you want to calculate $a^e \bmod n$

Sizes to right not unusual in RSA

a – 150 digits

e – 10^{120}, 121 digits

n – 150 digits

- 1^{st} try: Calculate a^e (how) and then take $\bmod n$

  No! a^e has $\sim 150 \cdot 10^{120}$ digits.

  It won't fit in our computer!

- 2^{nd} try: Iteratively calculate values between 0 and n using

  $a^{i+1} \bmod n = a\left(a^i \bmod n\right) \bmod n$

  No! Too many iterations.

  Sun would "burn out" before we finished!

Practical Aspects of Exponentiation $\bmod n$

Practical Aspects of Exponentiation $\bmod n$

Suppose you want to calculate $a^e \bmod n$

Sizes to right not unusual in RSA

a – 150 digits

e – 10^{120}, 121 digits

n – 150 digits

Practical Aspects of Exponentiation $\bmod n$

Suppose you want to calculate $a^e \bmod n$

Sizes to right not unusual in RSA

a – 150 digits

e – 10^{120}, 121 digits

n – 150 digits

- 3rd try: Use *Repeated Squaring*

Practical Aspects of Exponentiation $\bmod n$

Suppose you want to calculate $a^e \bmod n$

Sizes to right not unusual in RSA

a – 150 digits

e – 10^{120}, 121 digits

n – 150 digits

- 3rd try: Use *Repeated Squaring*

Idea: $a^{50} = a^{32} \cdot a^{16} \cdot a^2$ so we could calculate a^{50} by

Practical Aspects of Exponentiation $\bmod n$

Suppose you want to calculate $a^e \bmod n$

Sizes to right not unusual in RSA

a – 150 digits

e – 10^{120}, 121 digits

n – 150 digits

- 3^{rd} try: Use *Repeated Squaring*

Idea: $a^{50} = a^{32} \cdot a^{16} \cdot a^2$ so we could calculate a^{50} by first using 5 muls to find

$$
\begin{aligned}
a^2 &= a \cdot a \\
a^4 &= a^2 \cdot a^2 \\
a^8 &= a^4 \cdot a^4 \\
a^{16} &= a^8 \cdot a^8 \\
a^{32} &= a^{16} \cdot a^{16}
\end{aligned}
$$

Practical Aspects of Exponentiation $\bmod n$

Suppose you want to calculate $a^e \bmod n$

Sizes to right not unusual in RSA

a – 150 digits

e – 10^{120}, 121 digits

n – 150 digits

- 3^{rd} try: Use *Repeated Squaring*

Idea: $a^{50} = a^{32} \cdot a^{16} \cdot a^2$ so we could calculate a^{50} by first using 5 muls to find

$$\begin{aligned} a^2 &= a \cdot a \\ a^4 &= a^2 \cdot a^2 \\ a^8 &= a^4 \cdot a^4 \\ a^{16} &= a^8 \cdot a^8 \\ a^{32} &= a^{16} \cdot a^{16} \end{aligned}$$

and then another 2 muls to get

$$\boxed{a^{50} = a^{32} \cdot a^{16} \cdot a^2}$$

Practical Aspects of Exponentiation $\bmod n$

Suppose you want to calculate $a^e \bmod n$

Sizes to right not unusual in RSA

a – 150 digits

e – 10^{120}, 121 digits

n – 150 digits

- 3^{rd} try: Use *Repeated Squaring*

Idea: $a^{50} = a^{32} \cdot a^{16} \cdot a^2$ so we could calculate a^{50} by

first using 5 muls to find

$$
\begin{aligned}
a^2 &= a \cdot a \\
a^4 &= a^2 \cdot a^2 \\
a^8 &= a^4 \cdot a^4 \\
a^{16} &= a^8 \cdot a^8 \\
a^{32} &= a^{16} \cdot a^{16}
\end{aligned}
$$

and then another 2 muls to get

$$\boxed{a^{50} = a^{32} \cdot a^{16} \cdot a^2}$$

Much better than 49 muls needed by iterative method!

Practical Aspects of Exponentiation $\bmod n$

Practical Aspects of Exponentiation $\bmod n$

Suppose you want to calculate $a^e \bmod n$

Sizes to right not unusual in RSA

a – 150 digits

e – 10^{120}, 121 digits

n – 150 digits

Practical Aspects of Exponentiation $\bmod n$

Suppose you want to calculate $a^e \bmod n$

Sizes to right not unusual in RSA

a – 150 digits

e – 10^{120}, 121 digits

n – 150 digits

- 3rd try: Use *Repeated Squaring*

Practical Aspects of Exponentiation $\bmod n$

Suppose you want to calculate $a^e \bmod n$

Sizes to right not unusual in RSA

a – 150 digits

e – 10^{120}, 121 digits

n – 150 digits

- 3^{rd} try: Use *Repeated Squaring* to calculate product $\bmod n$

Idea: $a^{50} = a^{32} \cdot a^{16} \cdot a^2$ so we could calculate $a^{50} \bmod n$ by

Practical Aspects of Exponentiation $\bmod n$

Suppose you want to calculate $a^e \bmod n$

Sizes to right not unusual in RSA

a – 150 digits

e – 10^{120}, 121 digits

n – 150 digits

- 3rd try: Use *Repeated Squaring* to calculate product $\bmod n$

Idea: $a^{50} = a^{32} \cdot a^{16} \cdot a^2$ so we could calculate $a^{50} \bmod n$ by

Setting $I_i = a^{2^i} \bmod n$ and

$$\begin{aligned}
I_1 &= (a \cdot a) \bmod n \\
I_2 &= (I_1 \cdot I_1) \bmod n \\
I_3 &= (I_2 \cdot I_2) \bmod n \\
I_4 &= (I_3 \cdot I_3) \bmod n \\
I_5 &= (I_4 \cdot I_4) \bmod n
\end{aligned}$$

Practical Aspects of Exponentiation $\bmod n$

Suppose you want to calculate $a^e \bmod n$

Sizes to right not unusual in RSA

a – 150 digits

e – 10^{120}, 121 digits

n – 150 digits

- 3^{rd} try: Use *Repeated Squaring* to calculate product $\bmod n$

Idea: $a^{50} = a^{32} \cdot a^{16} \cdot a^2$ so we could calculate $a^{50} \bmod n$ by

Setting $I_i = a^{2^i} \bmod n$ and

$$\begin{aligned}
I_1 &= (a \cdot a) \bmod n\\
I_2 &= (I_1 \cdot I_1) \bmod n\\
I_3 &= (I_2 \cdot I_2) \bmod n\\
I_4 &= (I_3 \cdot I_3) \bmod n\\
I_5 &= (I_4 \cdot I_4) \bmod n
\end{aligned}$$

and then calculating

$$\boxed{\begin{aligned} &a^{50} \bmod n = \\ &\quad (I_5 \cdot (I_4 \cdot I_1 \bmod n) \bmod n) \end{aligned}}$$

Practical Aspects of Exponentiation $\bmod n$

Suppose you want to calculate $a^e \bmod n$

Sizes to right not unusual in RSA

a – 150 digits

e – 10^{120}, 121 digits

n – 150 digits

- 3^{rd} try: Use *Repeated Squaring* to calculate product $\bmod n$

Idea: $a^{50} = a^{32} \cdot a^{16} \cdot a^2$ so we could calculate $a^{50} \bmod n$ by

Setting $I_i = a^{2^i} \bmod n$ and

$$
\begin{aligned}
I_1 &= (a \cdot a) \bmod n \\
I_2 &= (I_1 \cdot I_1) \bmod n \\
I_3 &= (I_2 \cdot I_2) \bmod n \\
I_4 &= (I_3 \cdot I_3) \bmod n \\
I_5 &= (I_4 \cdot I_4) \bmod n
\end{aligned}
$$

and then calculating

$$\boxed{\begin{aligned} &a^{50} \bmod n = \\ &\quad (I_5 \cdot (I_4 \cdot I_1 \bmod n) \bmod n) \end{aligned}}$$

Note: No factor is ever $\geq n$

Repeated squaring to evaluate $a^e \bmod n$

Repeated squaring to evaluate $a^e \bmod n$

- Calculate binary representation of e: $e_s e_{s-1} \cdots e_2 e_1 e_0$
  $e = \sum_{i=0}^{s} e_i 2^i$ and $s \leq \log_2 n$
  Example: $50 = 110010$

Repeated squaring to evaluate $a^e \bmod n$

- Calculate binary representation of e: $e_s e_{s-1} \cdots e_2 e_1 e_0$
  $e = \sum_{i=0}^{s} e_i 2^i$ and $s \le \log_2 n$
  Example: $50 = 110010$

- Now find $k_1, k_2, \ldots, k_t$ so that $e = 2^{k_1} + 2^{k_2} + \cdots + 2^{k_t}$.
  The k_i are just the locations of the 1s in the bin rep of e
  Example: $50 = 2^1 + 2^4 + 2^5$ so $(k_1, k_2, k_3) = (1, 4, 5)$

Repeated squaring to evaluate $a^e \bmod n$

- Calculate binary representation of e: $e_s e_{s-1} \cdots e_2 e_1 e_0$
  $e = \sum_{i=0}^{s} e_i 2^i$ and $s \leq \log_2 n$
  Example: $50 = 110010$

- Now find $k_1, k_2, \ldots, k_t$ so that $e = 2^{k_1} + 2^{k_2} + \cdots + 2^{k_t}$.
  The k_i are just the locations of the 1s in the bin rep of e
  Example: $50 = 2^1 + 2^4 + 2^5$ so $(k_1, k_2, k_3) = (1, 4, 5)$

- Calculate $I_0 = a$, $I_1 = (I_0)^2 \bmod n$, $I_2 = (I_1)^2 \bmod n$, $I_3 = (I_2)^2 \bmod n$, $\ldots$
  where $I_i = (I_{i-1})^2 \bmod n$ for $i = 1, 2, 3, \ldots, n$

Repeated squaring to evaluate $a^e \bmod n$

- Calculate binary representation of e: $e_s e_{s-1} \cdots e_2 e_1 e_0$
  $e = \sum_{i=0}^{s} e_i 2^i$ and $s \le \log_2 n$
  Example: $50 = 110010$

- Now find $k_1, k_2, \ldots, k_t$ so that $e = 2^{k_1} + 2^{k_2} + \cdots + 2^{k_t}$.
  The k_i are just the locations of the 1s in the bin rep of e
  Example: $50 = 2^1 + 2^4 + 2^5$ so $(k_1, k_2, k_3) = (1, 4, 5)$

- Calculate $I_0 = a$, $I_1 = (I_0)^2 \bmod n$, $I_2 = (I_1)^2 \bmod n$, $I_3 = (I_2)^2 \bmod n$, $\ldots$
  where $I_i = (I_{i-1})^2 \bmod n$ for $i = 1, 2, 3, \ldots, n$

- $a^e \bmod n = (I_{k_1} I_{k_2} \cdots I_{k_t}) \bmod n$ so we can calculate this using $t-1$ multiplications where no factor is ever $\ge n$.

- How many multiplications and mods does this procedure use to calculate $a^e \bmod n$?

- How many multiplications and mods does this procedure use to calculate $a^e \bmod n$?

- Note that if e has binary representation $e_s e_{s-1} \cdots e_2 e_1 e_0$ then it performs s multiplications and mods in the *repeated squaring* part, and, at most, another s multiplications and mods in the second part.

  Since $s \sim \log_2 e$ this means it performs at most around $2 \log_2 e \leq 2 \log_2 n$ of these operations.

  Compare this to the $e - 1$ operations it would require if we did naive exponentiation without repeated squaring.

- How many multiplications and mods does this procedure use to calculate $a^e \bmod n$?

- Note that if e has binary representation $e_s e_{s-1} \cdots e_2 e_1 e_0$ then it performs s multiplications and mods in the *repeated squaring* part, and, at most, another s multiplications and mods in the second part.

  Since $s \sim \log_2 e$ this means it performs at most around $2 \log_2 e \leq 2 \log_2 n$ of these operations.

  Compare this to the $e - 1$ operations it would require if we did naive exponentiation without repeated squaring.

- To put this in perspective, consider $e = 10^{120}$. This number is so big that, at current computer speeds, we would not be able to finish running the naive algorithm **before the sun died**. On the other hand, $2 \log_2 e = 240 \log_2 10 \sim 796$ so we could run the repeated squaring algorithm in **just a few seconds**!

Comment: This idea of designing *efficient* programs for solving problems and then analyzing their running times is something that you will see a lot more of in Data Structures and The Design and Analysis of Algorithms

2.3 The RSA Cryptosystem

- Assorted Tools and Definitions
- Exponentiation mod n
- The Rules of Exponents
- Fermat's Little Theorem
- The RSA Cryptosystem
- Practical Aspects of Exponentiation mod n
- The Chinese Remainder Theorem

The Chinese Remainder Theorem

While proving the correctness of RSA, we proved the following:

If (i) $0 \le x < n = pq$,
(ii) $x^{ed} \bmod p = x \bmod p$ and
(iii) $x^{ed} \bmod q = x \bmod q$

$$\Rightarrow \quad x^{ed} \pmod{n} = x$$

The Chinese Remainder Theorem

While proving the correctness of RSA, we proved the following:

If (i) $0 \le x < n = pq$,
(ii) $x^{ed} \bmod p = x \bmod p$ and
(iii) $x^{ed} \bmod q = x \bmod q$

$$\Rightarrow \quad x^{ed} \pmod{n} = x$$

This turns out to be a special case of a general rule:

The Chinese Remainder Theorem

The Chinese Remainder Theorem

The Chinese Remainder Theorem

For each $x \in Z_{15}$, write
$x \bmod 3$ and $x \bmod 5$.
Is x uniquely determined
by these values?

The Chinese Remainder Theorem

For each $x \in Z_{15}$, write $x \bmod 3$ and $x \bmod 5$.

Is x uniquely determined by these values?

x	x mod 3	x mod 5
0	0	0
1	1	1
2	2	2
3	0	3
4	1	4
5	2	0
6	0	1
7	1	2
8	2	3
9	0	4
10	1	0
11	2	1
12	0	2
13	1	3
14	2	4

The Chinese Remainder Theorem

For each $x \in Z_{15}$, write $x \bmod 3$ and $x \bmod 5$.

Is x uniquely determined by these values? Yes!

Each $x \in Z_{15}$ has a different $x \bmod 3$, $x \bmod 5$ pair.

x	x mod 3	x mod 5
0	0	0
1	1	1
2	2	2
3	0	3
4	1	4
5	2	0
6	0	1
7	1	2
8	2	3
9	0	4
10	1	0
11	2	1
12	0	2
13	1	3
14	2	4

The Chinese Remainder Theorem

For each $x \in Z_{15}$, write $x \bmod 3$ and $x \bmod 5$.

Is x uniquely determined by these values? Yes!

Each $x \in Z_{15}$ has a different $x \bmod 3$, $x \bmod 5$ pair.

Thus, the function $f(x) = (x \bmod 3,\ x \bmod 5)$ from Z_{15} to the 15 pairs (i, j) with $0 \le i < 3$ and $0 \le j < 5$ is one-to-one.

x	x mod 3	x mod 5
0	0	0
1	1	1
2	2	2
3	0	3
4	1	4
5	2	0
6	0	1
7	1	2
8	2	3
9	0	4
10	1	0
11	2	1
12	0	2
13	1	3
14	2	4

The Chinese Remainder Theorem

For each $x \in Z_{15}$, write $x \bmod 3$ and $x \bmod 5$.
Is x uniquely determined by these values? Yes!

Each $x \in Z_{15}$ has a different $x \bmod 3$, $x \bmod 5$ pair.

Thus, the function $f(x) = (x \bmod 3,\ x \bmod 5)$ from Z_{15} to the 15 pairs (i, j) with $0 \le i < 3$ and $0 \le j < 5$ is one-to-one.

$\Rightarrow$ x is uniquely determined by its pair of remainders.

x	x mod 3	x mod 5
0	0	0
1	1	1
2	2	2
3	0	3
4	1	4
5	2	0
6	0	1
7	1	2
8	2	3
9	0	4
10	1	0
11	2	1
12	0	2
13	1	3
14	2	4

Theorem 2.24 (Chinese Remainder Theorem)
If m and n are relatively prime integers, then the equations $x \bmod m = a \in Z_m$ and $x \bmod n = b \in Z_n$ have one and only one solution for an integer x between 0 and $mn - 1$.

<table><tr><td>

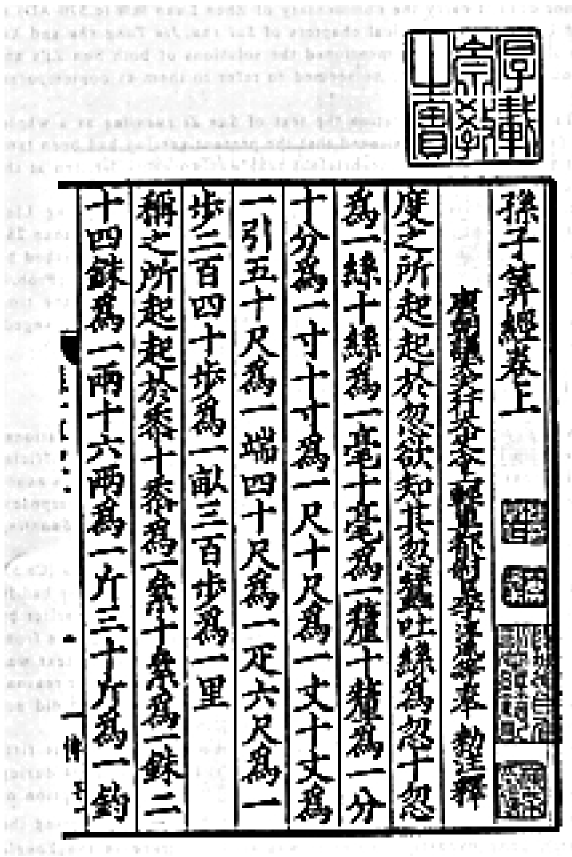
Theorem 2.24 (Chinese Remainder Theorem)

If m and n are relatively prime integers, then the equations $x \bmod m = a \in Z_m$ and $x \bmod n = b \in Z_n$ have one and only one solution for an integer x between 0 and $mn - 1$.

</td></tr></table>

Why is this called the Chinese Remainder Theorem?

The earliest reference known is from the Sun Tzu Suan Ching (also known as Sunzi Suanjing) written in approximately the late third century by Sun Zi. Problem 26 in the third volume of the Sun Tzu Suan Ching offers the earliest recorded description of the Chinese Remainder Problem.



孫子算經卷上

度之所起起於忽欲知其忽蠶吐絲為忽十忽
為一絲十絲為一毫十毫為一氂十氂為一分
十分為一寸十寸為一尺十尺為一丈十丈為
一引五十尺為一端四十尺為一疋六尺為一
步二百四十步為一畝三百步為一里
稱之所起起於黍十黍為一絫十絫為一銖二
十四銖為一兩十六兩為一斤三十斤為一鈞

see *http://www.math.sfu.ca/histmath/China/3rdCenturyBC/CRP1.html* for more details

Theorem 2.24 (Chinese Remainder Theorem)
If m and n are relatively prime integers, then the equations $x \bmod m = a \in Z_m$ and $x \bmod n = b \in Z_n$ have one and only one solution for an integer x between 0 and $mn - 1$.

Theorem 2.24 (Chinese Remainder Theorem)
If m and n are relatively prime integers, then the equations $x \bmod m = a \in Z_m$ and $x \bmod n = b \in Z_n$ have one and only one solution for an integer x between 0 and $mn - 1$.

Proof: Let $f(x) = (x \bmod m,\ x \bmod n)$

$f : \{0, 1, 2, \ldots, mn-1\} \rightarrow$ the pairs (a, b) : $0 \leq a < m$ and $0 \leq b < n$

Theorem 2.24 (Chinese Remainder Theorem)
If m and n are relatively prime integers, then the equations $x \bmod m = a \in Z_m$ and $x \bmod n = b \in Z_n$ have one and only one solution for an integer x between 0 and $mn - 1$.

Proof: Let $f(x) = (x \bmod m,\, x \bmod n)$

$f : \{0, 1, 2, \ldots, mn - 1\} \rightarrow$ the pairs $(a, b) :\ 0 \leq a < m$ and $0 \leq b < n$

To prove the theorem we must show that f is a bijection.
f is mapping a set of size mn to a set of size mn, so,
to prove it's a bijection, it's enough to prove that f is onto.

Theorem 2.24 (Chinese Remainder Theorem)
If m and n are relatively prime integers, then the equations $x \bmod m = a \in Z_m$ and $x \bmod n = b \in Z_n$ have one and only one solution for an integer x between 0 and $mn - 1$.

Proof: Let $f(x) = (x \bmod m,\, x \bmod n)$

$f : \{0, 1, 2, \ldots, mn - 1\} \rightarrow$ the pairs $(a, b) :\ 0 \leq a < m$ and $0 \leq b < n$

To prove the theorem we must show that f is a bijection.
f is mapping a set of size mn to a set of size mn, so,
to prove it's a bijection, it's enough to prove that f is onto.

Given (a, b) we will now see how to construct a y s.t.
$y \bmod m = a$ and $y \bmod n = b$

Theorem 2.24 (Chinese Remainder Theorem)
If m and n are relatively prime integers, then the equations $x \bmod m = a \in Z_m$ and $x \bmod n = b \in Z_n$ have one and only one solution for an integer x between 0 and $mn - 1$.

Proof: Let $f(x) = (x \bmod m,\, x \bmod n)$

$f : \{0, 1, 2, \ldots, mn - 1\} \rightarrow$ the pairs $(a, b):\ 0 \le a < m$ and $0 \le b < n$

To prove the theorem we must show that f is a bijection.
f is mapping a set of size mn to a set of size mn, so,
to prove it's a bijection, it's enough to prove that f is onto.

Given (a, b) we will now see how to construct a y s.t.
$y \bmod m = a$ and $y \bmod n = b$

y might not be $< mn$ but we can set $x = y \bmod (mn)$ and get
$x < mn$ and (why?) $x \bmod m = a$ and $x \bmod n = b$

Theorem 2.24 (Chinese Remainder Theorem) If m and n are relatively prime integers, then the equations $x \bmod m = a \in Z_m$ and $x \bmod n = b \in Z_n$ have one and only one solution for an integer x between 0 and $mn - 1$.

Proof: (cont) Let $f(x) = (x \bmod m,\ x \bmod n)$

Theorem 2.24 (Chinese Remainder Theorem)
If m and n are relatively prime integers, then the equations $x \bmod m = a \in Z_m$ and $x \bmod n = b \in Z_n$ have one and only one solution for an integer x between 0 and $mn - 1$.

Proof: (cont) Let $f(x) = (x \bmod m, x \bmod n)$

Given (a, b) want y s.t. $y \bmod m = a$ and $y \bmod n = b$

Theorem 2.24 (Chinese Remainder Theorem)
If m and n are relatively prime integers, then the equations $x \bmod m = a \in Z_m$ and $x \bmod n = b \in Z_n$ have one and only one solution for an integer x between 0 and $mn - 1$.

Proof: (cont) Let $f(x) = (x \bmod m,\ x \bmod n)$

Given (a, b) want y s.t. $y \bmod m = a$ and $y \bmod n = b$

- Since $gcd(m, n) = 1$ there exists, $\overline{m}$ s.t. $m \cdot \overline{m} = 1 \bmod n$

Theorem 2.24 (Chinese Remainder Theorem)
If m and n are relatively prime integers, then the equations $x \bmod m = a \in Z_m$ and $x \bmod n = b \in Z_n$ have one and only one solution for an integer x between 0 and $mn - 1$.

Proof: (cont) Let $f(x) = (x \bmod m, x \bmod n)$

Given (a, b) want y s.t. $y \bmod m = a$ and $y \bmod n = b$

- Since $gcd(m, n) = 1$ there exists, $\overline{m}$ s.t. $m \cdot \overline{m} = 1 \bmod n$
- Similarly there exists, $\overline{n}$ s.t. $n \cdot \overline{n} = 1 \bmod m$

Theorem 2.24 (Chinese Remainder Theorem)
If m and n are relatively prime integers, then the equations $x \bmod m = a \in Z_m$ and $x \bmod n = b \in Z_n$ have one and only one solution for an integer x between 0 and $mn - 1$.

Proof: (cont) Let $f(x) = (x \bmod m, x \bmod n)$

Given (a, b) want y s.t. $y \bmod m = a$ and $y \bmod n = b$

- Since $gcd(m, n) = 1$ there exists, $\overline{m}$ s.t. $m \cdot \overline{m} = 1 \bmod n$
- Similarly there exists, $\overline{n}$ s.t. $n \cdot \overline{n} = 1 \bmod m$
- Set $y = a\overline{n}n + b\overline{m}m$

Theorem 2.24 (Chinese Remainder Theorem)
If m and n are relatively prime integers, then the equations $x \bmod m = a \in Z_m$ and $x \bmod n = b \in Z_n$ have one and only one solution for an integer x between 0 and $mn - 1$.

Proof: (cont) Let $f(x) = (x \bmod m,\ x \bmod n)$

Given (a, b) want y s.t. $y \bmod m = a$ and $y \bmod n = b$

- Since $gcd(m, n) = 1$ there exists, $\overline{m}$ s.t. $m \cdot \overline{m} = 1 \bmod n$
- Similarly there exists, $\overline{n}$ s.t. $n \cdot \overline{n} = 1 \bmod m$
- Set $y = a\overline{n}n + b\overline{m}m$
- Then $$\begin{aligned} y \bmod m &= (a\overline{n}n) \bmod m = a \\ y \bmod n &= (b\overline{m}m) \bmod n = b \end{aligned}$$

Theorem 2.24 (Chinese Remainder Theorem)
If m and n are relatively prime integers, then the equations $x \bmod m = a \in Z_m$ and $x \bmod n = b \in Z_n$ have one and only one solution for an integer x between 0 and $mn - 1$.

Proof: (cont) Let $f(x) = (x \bmod m,\ x \bmod n)$

Given (a, b) want y s.t. $y \bmod m = a$ and $y \bmod n = b$

- Since $gcd(m, n) = 1$ there exists, $\overline{m}$ s.t. $m \cdot \overline{m} = 1 \bmod n$
- Similarly there exists, $\overline{n}$ s.t. $n \cdot \overline{n} = 1 \bmod m$
- Set $y = a\overline{n}n + b\overline{m}m$
- Then $\boxed{\begin{aligned} y \bmod m &= (a\overline{n}n) \bmod m = a \\ y \bmod n &= (b\overline{m}m) \bmod n = b \end{aligned}}$ $\leftarrow$ done!

Theorem 2.24 (Chinese Remainder Theorem)
If m and n are relatively prime integers, then the equations $x \bmod m = a \in Z_m$ and $x \bmod n = b \in Z_n$ have one and only one solution for an integer x between 0 and $mn - 1$.

Example: $m = 6$, $n = 11$, $a = 3$, $b = 7$

Theorem 2.24 (Chinese Remainder Theorem)
If m and n are relatively prime integers, then the equations $x \bmod m = a \in Z_m$ and $x \bmod n = b \in Z_n$ have one and only one solution for an integer x between 0 and $mn - 1$.

Example: $m = 6$, $n = 11$, $a = 3$, $b = 7$

- $\overline{m} = 2$ and $\overline{n} = 5$ since

$$6 \cdot 2 \bmod 11 \;=\; 12 \bmod 11 \;=\; 1$$
$$11 \cdot 5 \bmod 6 \;=\; 55 \bmod 6 \;=\; 1$$

Theorem 2.24 (Chinese Remainder Theorem)
If m and n are relatively prime integers, then the equations $x \bmod m = a \in Z_m$ and $x \bmod n = b \in Z_n$ have one and only one solution for an integer x between 0 and $mn - 1$.

Example: $m = 6, n = 11, a = 3, b = 7$

- $\overline{m} = 2$ and $\overline{n} = 5$ since

  $$6 \cdot 2 \bmod 11 \;=\; 12 \bmod 11 \;=\; 1$$
  $$11 \cdot 5 \bmod 6 \;=\; 55 \bmod 6 \;=\; 1$$

- Set $y = a\overline{n}n + b\overline{m}m = 3 \cdot 5 \cdot 11 + 7 \cdot 2 \cdot 6 = 249$

Theorem 2.24 (Chinese Remainder Theorem)
If m and n are relatively prime integers, then the equations $x \bmod m = a \in Z_m$ and $x \bmod n = b \in Z_n$ have one and only one solution for an integer x between 0 and $mn - 1$.

Example: $m = 6$, $n = 11$, $a = 3$, $b = 7$

- $\overline{m} = 2$ and $\overline{n} = 5$ since

  $$6 \cdot 2 \bmod 11 \;=\; 12 \bmod 11 \;=\; 1$$
  $$11 \cdot 5 \bmod 6 \;=\; 55 \bmod 6 \;=\; 1$$

- Set $y = a\overline{n}n + b\overline{m}m = 3 \cdot 5 \cdot 11 + 7 \cdot 2 \cdot 6 = 249$

- $249 = 3 * 66 + 51$ so $x = y \bmod (nm) = 51$

Theorem 2.24 (Chinese Remainder Theorem)
If m and n are relatively prime integers, then the equations $x \bmod m = a \in Z_m$ and $x \bmod n = b \in Z_n$ have one and only one solution for an integer x between 0 and $mn - 1$.

Example: $m = 6$, $n = 11$, $a = 3$, $b = 7$

- $\overline{m} = 2$ and $\overline{n} = 5$ since

  $$6 \cdot 2 \bmod 11 \;=\; 12 \bmod 11 \;=\; 1$$
  $$11 \cdot 5 \bmod 6 \;=\; 55 \bmod 6 \;=\; 1$$

- Set $y = a\overline{n}n + b\overline{m}m = 3 \cdot 5 \cdot 11 + 7 \cdot 2 \cdot 6 = 249$

- $249 = 3 * 66 + 51$ so $x = y \bmod (nm) = 51$

- Reality Check: $\boxed{\begin{array}{c} 51 \bmod 6 = 3 \\ 51 \bmod 11 = 7 \end{array}}$

Theorem 2.24 (Chinese Remainder Theorem)
If m and n are relatively prime integers, then the equations $x \bmod m = a \in Z_m$ and $x \bmod n = b \in Z_n$ have one and only one solution for an integer x between 0 and $mn - 1$.

Theorem 2.24 (Chinese Remainder Theorem)
If m and n are relatively prime integers, then the equations $x \bmod m = a \in Z_m$ and $x \bmod n = b \in Z_n$ have one and only one solution for an integer x between 0 and $mn - 1$.

- That proof is an example of why I can't do math!
  It was magically pulled out of thin air!
  Impossible to do myself!

Theorem 2.24 (Chinese Remainder Theorem)
If m and n are relatively prime integers, then the equations $x \bmod m = a \in Z_m$ and $x \bmod n = b \in Z_n$ have one and only one solution for an integer x between 0 and $mn - 1$.

- That proof is an example of why I can't do math!
  It was magically pulled out of thin air!
  Impossible to do myself!

- It's not magic. You had all of the pieces already.

Theorem 2.24 (Chinese Remainder Theorem)
If m and n are relatively prime integers, then the equations $x \bmod m = a \in Z_m$ and $x \bmod n = b \in Z_n$ have one and only one solution for an integer x between 0 and $mn - 1$.

- That proof is an example of why I can't do math!
  It was magically pulled out of thin air!
  Impossible to do myself!
- It's not magic. You had all of the pieces already.
- Think first. You want to build y such that
  $y \bmod m = a$ and $y \bmod n = b$

Theorem 2.24 (Chinese Remainder Theorem)
If m and n are relatively prime integers, then the equations $x \bmod m = a \in Z_m$ and $x \bmod n = b \in Z_n$ have one and only one solution for an integer x between 0 and $mn-1$.

- That proof is an example of why I can't do math!
  It was magically pulled out of thin air!
  Impossible to do myself!

- It's not magic. You had all of the pieces already.

- Think first. You want to build y such that
  $y \bmod m = a$ and $y \bmod n = b$

- Suppose you knew α, β such that
  $\alpha \bmod m = 1,\ \alpha \bmod n = 0, \quad \beta \bmod n = 1,\ \beta \bmod m = 0$
  $\Rightarrow \quad y = a\alpha + b\beta$ satisfies our requirements

Theorem 2.24 (Chinese Remainder Theorem)
If m and n are relatively prime integers, then the equations $x \bmod m = a \in Z_m$ and $x \bmod n = b \in Z_n$ have one and only one solution for an integer x between 0 and $mn - 1$.

- That proof is an example of why I can't do math!
  It was magically pulled out of thin air!
  Impossible to do myself!

- It's not magic. You had all of the pieces already.

- Think first. You want to build y such that
  $y \bmod m = a$ and $y \bmod n = b$

- Suppose you knew α, β such that
  $\alpha \bmod m = 1, \ \alpha \bmod n = 0, \quad \beta \bmod n = 1, \ \beta \bmod m = 0$

  $\Rightarrow \quad y = a\alpha + b\beta$ satisfies our requirements

- such an α must be a multiple of n. But $\alpha \bmod m = 1$ then implies that $\alpha = \overline{n}n$ where $\overline{n}n \bmod m = 1$. Similarly $\beta = \overline{m}m$

Theorem 2.24 (Chinese Remainder Theorem)
If m and n are relatively prime integers, then the equations $x \bmod m = a \in Z_m$ and $x \bmod n = b \in Z_n$ have one and only one solution for an integer x between 0 and $mn - 1$.

- That proof is an example of why I can't do math!
  It was magically pulled out of thin air!
  Impossible to do myself!

- It's not magic. You had all of the pieces already.

- Think first. You want to build y such that
  $y \bmod m = a$ and $y \bmod n = b$

- Suppose you knew α, β such that
  $\alpha \bmod m = 1, \ \alpha \bmod n = 0, \quad \beta \bmod n = 1, \ \beta \bmod m = 0$

  $\Rightarrow \quad y = a\alpha + b\beta$ satisfies our requirements

- such an α must be a multiple of n. But $\alpha \bmod m = 1$ then implies that $\alpha = \overline{n}n$ where $\overline{n}n \bmod m = 1$. Similarly $\beta = \overline{m}m$

  $\Rightarrow \quad y = a\overline{n}n + b\overline{m}m$

- The textbook gives a different proof than our's.

- The textbook gives a different proof than our's.
- The book's proof, which you should also read, uses proof by contradiction.

- The textbook gives a different proof than our's.
- The book's proof, which you should also read, uses proof by contradiction.
- Our proof was a constructive proof.

  We not only showed that the theorem was correct, but we did so by giving a procedure to construct an x satisfying the statement of the theorem.