

Today

Today

Finish Euclid.

Today

Finish Euclid.

Bijection/CRT/Isomorphism.

Today

Finish Euclid.

Bijection/CRT/Isomorphism.

Fermat's Little Theorem.

Finding an inverse?

We showed how to efficiently tell if there is an inverse.

Finding an inverse?

We showed how to efficiently tell if there is an inverse.

Extend euclid to find inverse.

Euclid's GCD algorithm.

```
(define (euclid x y)
  (if (= y 0)
      x
      (euclid y (mod x y))))
```

Euclid's GCD algorithm.

```
(define (euclid x y)
  (if (= y 0)
      x
      (euclid y (mod x y))))
```

Computes the $\text{gcd}(x,y)$ in $O(n)$ divisions. (Remember $n = \log_2 x$.)

Euclid's GCD algorithm.

```
(define (euclid x y)
  (if (= y 0)
      x
      (euclid y (mod x y))))
```

Computes the $\text{gcd}(x, y)$ in $O(n)$ divisions. (Remember $n = \log_2 x$.)

For x and m , if $\text{gcd}(x, m) = 1$ then x has an inverse modulo m .

Multiplicative Inverse.

GCD algorithm used to tell **if** there is a multiplicative inverse.

Multiplicative Inverse.

GCD algorithm used to tell **if** there is a multiplicative inverse.

How do we **find** a multiplicative inverse?

Extended GCD

Euclid's Extended GCD Theorem: For any x, y there are integers a, b such that

$$ax + by$$

Extended GCD

Euclid's Extended GCD Theorem: For any x, y there are integers a, b such that

$$ax + by = d \quad \text{where } d = \gcd(x, y).$$

Extended GCD

Euclid's Extended GCD Theorem: For any x, y there are integers a, b such that

$$ax + by = d \quad \text{where } d = \gcd(x, y).$$

“Make d out of sum of multiples of x and y .”

Extended GCD

Euclid's Extended GCD Theorem: For any x, y there are integers a, b such that

$$ax + by = d \quad \text{where } d = \gcd(x, y).$$

“Make d out of sum of multiples of x and y .”

What is multiplicative inverse of x modulo m ?

Extended GCD

Euclid's Extended GCD Theorem: For any x, y there are integers a, b such that

$$ax + by = d \quad \text{where } d = \gcd(x, y).$$

“Make d out of sum of multiples of x and y .”

What is multiplicative inverse of x modulo m ?

By extended GCD theorem, when $\gcd(x, m) = 1$.

Extended GCD

Euclid's Extended GCD Theorem: For any x, y there are integers a, b such that

$$ax + by = d \quad \text{where } d = \gcd(x, y).$$

“Make d out of sum of multiples of x and y .”

What is multiplicative inverse of x modulo m ?

By extended GCD theorem, when $\gcd(x, m) = 1$.

$$ax + bm = 1$$

Extended GCD

Euclid's Extended GCD Theorem: For any x, y there are integers a, b such that

$$ax + by = d \quad \text{where } d = \gcd(x, y).$$

“Make d out of sum of multiples of x and y .”

What is multiplicative inverse of x modulo m ?

By extended GCD theorem, when $\gcd(x, m) = 1$.

$$\begin{aligned} ax + bm &= 1 \\ ax &\equiv 1 - bm \equiv 1 \pmod{m}. \end{aligned}$$

Extended GCD

Euclid's Extended GCD Theorem: For any x, y there are integers a, b such that

$$ax + by = d \quad \text{where } d = \gcd(x, y).$$

“Make d out of sum of multiples of x and y .”

What is multiplicative inverse of x modulo m ?

By extended GCD theorem, when $\gcd(x, m) = 1$.

$$\begin{aligned} ax + bm &= 1 \\ ax &\equiv 1 - bm \equiv 1 \pmod{m}. \end{aligned}$$

So a multiplicative inverse of $x \pmod{m}$!!

Extended GCD

Euclid's Extended GCD Theorem: For any x, y there are integers a, b such that

$$ax + by = d \quad \text{where } d = \gcd(x, y).$$

“Make d out of sum of multiples of x and y .”

What is multiplicative inverse of x modulo m ?

By extended GCD theorem, when $\gcd(x, m) = 1$.

$$\begin{aligned} ax + bm &= 1 \\ ax &\equiv 1 - bm \equiv 1 \pmod{m}. \end{aligned}$$

So a multiplicative inverse of $x \pmod{m}$!!

Example: For $x = 12$ and $y = 35$, $\gcd(12, 35) = 1$.

Extended GCD

Euclid's Extended GCD Theorem: For any x, y there are integers a, b such that

$$ax + by = d \quad \text{where } d = \gcd(x, y).$$

“Make d out of sum of multiples of x and y .”

What is multiplicative inverse of x modulo m ?

By extended GCD theorem, when $\gcd(x, m) = 1$.

$$\begin{aligned} ax + bm &= 1 \\ ax &\equiv 1 - bm \equiv 1 \pmod{m}. \end{aligned}$$

So a multiplicative inverse of $x \pmod{m}$!!

Example: For $x = 12$ and $y = 35$, $\gcd(12, 35) = 1$.

$$(3)12 + (-1)35 = 1.$$

Extended GCD

Euclid's Extended GCD Theorem: For any x, y there are integers a, b such that

$$ax + by = d \quad \text{where } d = \gcd(x, y).$$

“Make d out of sum of multiples of x and y .”

What is multiplicative inverse of x modulo m ?

By extended GCD theorem, when $\gcd(x, m) = 1$.

$$\begin{aligned} ax + bm &= 1 \\ ax &\equiv 1 - bm \equiv 1 \pmod{m}. \end{aligned}$$

So a multiplicative inverse of $x \pmod{m}$!!

Example: For $x = 12$ and $y = 35$, $\gcd(12, 35) = 1$.

$$(3)12 + (-1)35 = 1.$$

$$a = 3 \text{ and } b = -1.$$

Extended GCD

Euclid's Extended GCD Theorem: For any x, y there are integers a, b such that

$$ax + by = d \quad \text{where } d = \gcd(x, y).$$

“Make d out of sum of multiples of x and y .”

What is multiplicative inverse of x modulo m ?

By extended GCD theorem, when $\gcd(x, m) = 1$.

$$\begin{aligned} ax + bm &= 1 \\ ax &\equiv 1 - bm \equiv 1 \pmod{m}. \end{aligned}$$

So a multiplicative inverse of $x \pmod{m}$!!

Example: For $x = 12$ and $y = 35$, $\gcd(12, 35) = 1$.

$$(3)12 + (-1)35 = 1.$$

$$a = 3 \text{ and } b = -1.$$

The multiplicative inverse of $12 \pmod{35}$ is 3 .

Extended GCD

Euclid's Extended GCD Theorem: For any x, y there are integers a, b such that

$$ax + by = d \quad \text{where } d = \gcd(x, y).$$

“Make d out of sum of multiples of x and y .”

What is multiplicative inverse of x modulo m ?

By extended GCD theorem, when $\gcd(x, m) = 1$.

$$\begin{aligned} ax + bm &= 1 \\ ax &\equiv 1 - bm \equiv 1 \pmod{m}. \end{aligned}$$

So a multiplicative inverse of $x \pmod{m}$!!

Example: For $x = 12$ and $y = 35$, $\gcd(12, 35) = 1$.

$$(3)12 + (-1)35 = 1.$$

$$a = 3 \text{ and } b = -1.$$

The multiplicative inverse of $12 \pmod{35}$ is 3 .

Check: $3(12)$

Extended GCD

Euclid's Extended GCD Theorem: For any x, y there are integers a, b such that

$$ax + by = d \quad \text{where } d = \gcd(x, y).$$

“Make d out of sum of multiples of x and y .”

What is multiplicative inverse of x modulo m ?

By extended GCD theorem, when $\gcd(x, m) = 1$.

$$\begin{aligned} ax + bm &= 1 \\ ax &\equiv 1 - bm \equiv 1 \pmod{m}. \end{aligned}$$

So a multiplicative inverse of $x \pmod{m}$!!

Example: For $x = 12$ and $y = 35$, $\gcd(12, 35) = 1$.

$$(3)12 + (-1)35 = 1.$$

$$a = 3 \text{ and } b = -1.$$

The multiplicative inverse of $12 \pmod{35}$ is 3 .

Check: $3(12) = 36$

Extended GCD

Euclid's Extended GCD Theorem: For any x, y there are integers a, b such that

$$ax + by = d \quad \text{where } d = \gcd(x, y).$$

“Make d out of sum of multiples of x and y .”

What is multiplicative inverse of x modulo m ?

By extended GCD theorem, when $\gcd(x, m) = 1$.

$$\begin{aligned} ax + bm &= 1 \\ ax &\equiv 1 - bm \equiv 1 \pmod{m}. \end{aligned}$$

So a multiplicative inverse of $x \pmod{m}$!!

Example: For $x = 12$ and $y = 35$, $\gcd(12, 35) = 1$.

$$(3)12 + (-1)35 = 1.$$

$$a = 3 \text{ and } b = -1.$$

The multiplicative inverse of $12 \pmod{35}$ is 3 .

Check: $3(12) = 36 = 1 \pmod{35}$.

Extended GCD Algorithm.

```
ext-gcd(x,y)
  if y = 0 then return(x, 1, 0)
  else
    (d, a, b) := ext-gcd(y, mod(x,y))
    return (d, b, a - floor(x/y) * b)
```

Extended GCD Algorithm.

```
ext-gcd(x, y)
  if y = 0 then return(x, 1, 0)
  else
    (d, a, b) := ext-gcd(y, mod(x, y))
    return (d, b, a - floor(x/y) * b)
```

Claim: Returns (d, a, b) : $d = \gcd(x, y)$ and $d = ax + by$.

Extended GCD Algorithm.

```
ext-gcd(x, y)
  if y = 0 then return(x, 1, 0)
  else
    (d, a, b) := ext-gcd(y, mod(x, y))
    return (d, b, a - floor(x/y) * b)
```

Claim: Returns (d, a, b) : $d = \gcd(x, y)$ and $d = ax + by$.

Example:

```
ext-gcd(35, 12)
```

Extended GCD Algorithm.

```
ext-gcd(x, y)
  if y = 0 then return(x, 1, 0)
  else
    (d, a, b) := ext-gcd(y, mod(x, y))
    return (d, b, a - floor(x/y) * b)
```

Claim: Returns (d, a, b) : $d = \gcd(x, y)$ and $d = ax + by$.

Example:

```
ext-gcd(35, 12)
  ext-gcd(12, 11)
```

Extended GCD Algorithm.

```
ext-gcd(x, y)
  if y = 0 then return(x, 1, 0)
  else
    (d, a, b) := ext-gcd(y, mod(x, y))
    return (d, b, a - floor(x/y) * b)
```

Claim: Returns (d, a, b) : $d = \gcd(x, y)$ and $d = ax + by$.

Example:

```
ext-gcd(35, 12)
  ext-gcd(12, 11)
    ext-gcd(11, 1)
```

Extended GCD Algorithm.

```
ext-gcd(x, y)
  if y = 0 then return(x, 1, 0)
  else
    (d, a, b) := ext-gcd(y, mod(x, y))
    return (d, b, a - floor(x/y) * b)
```

Claim: Returns (d, a, b) : $d = \gcd(x, y)$ and $d = ax + by$.

Example:

```
ext-gcd(35, 12)
  ext-gcd(12, 11)
    ext-gcd(11, 1)
      ext-gcd(1, 0)
```


Extended GCD Algorithm.

```
ext-gcd(x,y)
  if y = 0 then return(x, 1, 0)
  else
    (d, a, b) := ext-gcd(y, mod(x,y))
    return (d, b, a - floor(x/y) * b)
```

Claim: Returns (d, a, b) : $d = \gcd(x, y)$ and $d = ax + by$.

Example: $a - \lfloor x/y \rfloor \cdot b =$

```
ext-gcd(35,12)
  ext-gcd(12, 11)
    ext-gcd(11, 1)
      ext-gcd(1,0)
        return (1,1,0) ;; 1 = (1)1 + (0) 0
```

Extended GCD Algorithm.

```
ext-gcd(x, y)
  if y = 0 then return(x, 1, 0)
  else
    (d, a, b) := ext-gcd(y, mod(x, y))
    return (d, b, a - floor(x/y) * b)
```

Claim: Returns (d, a, b) : $d = \gcd(x, y)$ and $d = ax + by$.

Example: $a - \lfloor x/y \rfloor \cdot b = 1 - \lfloor 11/1 \rfloor \cdot 0 = 1$

```
ext-gcd(35, 12)
  ext-gcd(12, 11)
    ext-gcd(11, 1)
      ext-gcd(1, 0)
        return (1, 1, 0) ;; 1 = (1)1 + (0) 0
      return (1, 0, 1)   ;; 1 = (0)11 + (1)1
```

Extended GCD Algorithm.

```
ext-gcd(x,y)
  if y = 0 then return(x, 1, 0)
  else
    (d, a, b) := ext-gcd(y, mod(x,y))
    return (d, b, a - floor(x/y) * b)
```

Claim: Returns (d, a, b) : $d = \gcd(x, y)$ and $d = ax + by$.

Example: $a - \lfloor x/y \rfloor \cdot b = 0 - \lfloor 12/11 \rfloor \cdot 1 = -1$

```
ext-gcd(35,12)
  ext-gcd(12, 11)
    ext-gcd(11, 1)
      ext-gcd(1,0)
        return (1,1,0) ;; 1 = (1)1 + (0) 0
      return (1,0,1)  ;; 1 = (0)11 + (1)1
    return (1,1,-1)  ;; 1 = (1)12 + (-1)11
```

Extended GCD Algorithm.

```
ext-gcd(x,y)
  if y = 0 then return(x, 1, 0)
  else
    (d, a, b) := ext-gcd(y, mod(x,y))
    return (d, b, a - floor(x/y) * b)
```

Claim: Returns (d, a, b) : $d = \gcd(x, y)$ and $d = ax + by$.

Example: $a - \lfloor x/y \rfloor \cdot b = 1 - \lfloor 35/12 \rfloor \cdot (-1) = 3$

```
ext-gcd(35,12)
  ext-gcd(12, 11)
    ext-gcd(11, 1)
      ext-gcd(1,0)
        return (1,1,0) ;; 1 = (1)1 + (0) 0
      return (1,0,1)   ;; 1 = (0)11 + (1)1
    return (1,1,-1)   ;; 1 = (1)12 + (-1)11
  return (1,-1, 3)   ;; 1 = (-1)35 + (3)12
```

Extended GCD Algorithm.

```
ext-gcd(x,y)
  if y = 0 then return(x, 1, 0)
  else
    (d, a, b) := ext-gcd(y, mod(x,y))
    return (d, b, a - floor(x/y) * b)
```

Claim: Returns (d, a, b) : $d = \gcd(x, y)$ and $d = ax + by$.

Example:

```
ext-gcd(35,12)
  ext-gcd(12, 11)
    ext-gcd(11, 1)
      ext-gcd(1,0)
        return (1,1,0) ;; 1 = (1)1 + (0) 0
      return (1,0,1)  ;; 1 = (0)11 + (1)1
    return (1,1,-1)  ;; 1 = (1)12 + (-1)11
  return (1,-1, 3)  ;; 1 = (-1)35 + (3)12
```

Extended GCD Algorithm.

```
ext-gcd(x,y)
  if y = 0 then return(x, 1, 0)
  else
    (d, a, b) := ext-gcd(y, mod(x,y))
    return (d, b, a - floor(x/y) * b)
```

Extended GCD Algorithm.

```
ext-gcd(x, y)
  if y = 0 then return(x, 1, 0)
  else
    (d, a, b) := ext-gcd(y, mod(x, y))
    return (d, b, a - floor(x/y) * b)
```

Theorem: Returns (d, a, b) , where $d = \gcd(x, y)$ and

$$d = ax + by.$$

Correctness.

Proof: Strong Induction.¹

¹Assume d is $\gcd(x, y)$ by previous proof.

Correctness.

Proof: Strong Induction.¹

Base: $\text{ext-gcd}(x, 0)$ returns $(d = x, 1, 0)$ with $x = (1)x + (0)y$.

¹Assume d is $\text{gcd}(x, y)$ by previous proof.

Correctness.

Proof: Strong Induction.¹

Base: $\text{ext-gcd}(x, 0)$ returns $(d = x, 1, 0)$ with $x = (1)x + (0)y$.

Induction Step: Returns (d, A, B) with $d = Ax + By$

¹Assume d is $\text{gcd}(x, y)$ by previous proof.

Correctness.

Proof: Strong Induction.¹

Base: $\text{ext-gcd}(x, 0)$ returns $(d = x, 1, 0)$ with $x = (1)x + (0)y$.

Induction Step: Returns (d, A, B) with $d = Ax + By$

Ind hyp: $\text{ext-gcd}(y, \text{ mod } (x, y))$ returns (d, a, b) with

$$d = ay + b(\text{ mod } (x, y))$$

¹Assume d is $\text{gcd}(x, y)$ by previous proof.

Correctness.

Proof: Strong Induction.¹

Base: $\text{ext-gcd}(x, 0)$ returns $(d = x, 1, 0)$ with $x = (1)x + (0)y$.

Induction Step: Returns (d, A, B) with $d = Ax + By$

Ind hyp: $\text{ext-gcd}(y, \text{ mod } (x, y))$ returns (d, a, b) with

$$d = ay + b(\text{ mod } (x, y))$$

$\text{ext-gcd}(x, y)$ calls $\text{ext-gcd}(y, \text{ mod } (x, y))$ so

¹Assume d is $\text{gcd}(x, y)$ by previous proof.

Correctness.

Proof: Strong Induction.¹

Base: $\text{ext-gcd}(x, 0)$ returns $(d = x, 1, 0)$ with $x = (1)x + (0)y$.

Induction Step: Returns (d, A, B) with $d = Ax + By$

Ind hyp: $\text{ext-gcd}(y, \text{ mod } (x, y))$ returns (d, a, b) with

$$d = ay + b(\text{ mod } (x, y))$$

$\text{ext-gcd}(x, y)$ calls $\text{ext-gcd}(y, \text{ mod } (x, y))$ so

$$d = ay + b \cdot (\text{ mod } (x, y))$$

¹Assume d is $\text{gcd}(x, y)$ by previous proof.

Correctness.

Proof: Strong Induction.¹

Base: $\text{ext-gcd}(x, 0)$ returns $(d = x, 1, 0)$ with $x = (1)x + (0)y$.

Induction Step: Returns (d, A, B) with $d = Ax + By$

Ind hyp: $\text{ext-gcd}(y, \text{ mod}(x, y))$ returns (d, a, b) with

$$d = ay + b(\text{ mod}(x, y))$$

$\text{ext-gcd}(x, y)$ calls $\text{ext-gcd}(y, \text{ mod}(x, y))$ so

$$\begin{aligned}d &= ay + b \cdot (\text{ mod}(x, y)) \\ &= ay + b \cdot (x - \lfloor \frac{x}{y} \rfloor y)\end{aligned}$$

¹Assume d is $\text{gcd}(x, y)$ by previous proof.

Correctness.

Proof: Strong Induction.¹

Base: $\text{ext-gcd}(x, 0)$ returns $(d = x, 1, 0)$ with $x = (1)x + (0)y$.

Induction Step: Returns (d, A, B) with $d = Ax + By$

Ind hyp: $\text{ext-gcd}(y, \text{ mod}(x, y))$ returns (d, a, b) with

$$d = ay + b(\text{ mod}(x, y))$$

$\text{ext-gcd}(x, y)$ calls $\text{ext-gcd}(y, \text{ mod}(x, y))$ so

$$\begin{aligned}d &= ay + b \cdot (\text{ mod}(x, y)) \\ &= ay + b \cdot (x - \lfloor \frac{x}{y} \rfloor y) \\ &= bx + (a - \lfloor \frac{x}{y} \rfloor \cdot b)y\end{aligned}$$

¹Assume d is $\text{gcd}(x, y)$ by previous proof.

Correctness.

Proof: Strong Induction.¹

Base: $\text{ext-gcd}(x, 0)$ returns $(d = x, 1, 0)$ with $x = (1)x + (0)y$.

Induction Step: Returns (d, A, B) with $d = Ax + By$

Ind hyp: $\text{ext-gcd}(y, \text{ mod}(x, y))$ returns (d, a, b) with

$$d = ay + b(\text{ mod}(x, y))$$

$\text{ext-gcd}(x, y)$ calls $\text{ext-gcd}(y, \text{ mod}(x, y))$ so

$$\begin{aligned}d &= ay + b \cdot (\text{ mod}(x, y)) \\ &= ay + b \cdot (x - \lfloor \frac{x}{y} \rfloor y) \\ &= bx + (a - \lfloor \frac{x}{y} \rfloor \cdot b)y\end{aligned}$$

And ext-gcd returns $(d, b, (a - \lfloor \frac{x}{y} \rfloor \cdot b))$ so theorem holds!

¹Assume d is $\text{gcd}(x, y)$ by previous proof.

Correctness.

Proof: Strong Induction.¹

Base: $\text{ext-gcd}(x, 0)$ returns $(d = x, 1, 0)$ with $x = (1)x + (0)y$.

Induction Step: Returns (d, A, B) with $d = Ax + By$

Ind hyp: $\text{ext-gcd}(y, \text{ mod}(x, y))$ returns (d, a, b) with

$$d = ay + b(\text{ mod}(x, y))$$

$\text{ext-gcd}(x, y)$ calls $\text{ext-gcd}(y, \text{ mod}(x, y))$ so

$$\begin{aligned}d &= ay + b \cdot (\text{ mod}(x, y)) \\ &= ay + b \cdot (x - \lfloor \frac{x}{y} \rfloor y) \\ &= bx + (a - \lfloor \frac{x}{y} \rfloor \cdot b)y\end{aligned}$$

And ext-gcd returns $(d, b, (a - \lfloor \frac{x}{y} \rfloor \cdot b))$ so theorem holds! □

¹Assume d is $\text{gcd}(x, y)$ by previous proof.

Review Proof: step.

```
ext-gcd(x, y)
  if y = 0 then return(x, 1, 0)
  else
    (d, a, b) := ext-gcd(y, mod(x, y))
    return (d, b, a - floor(x/y) * b)
```

Review Proof: step.

```
ext-gcd(x, y)
  if y = 0 then return(x, 1, 0)
  else
    (d, a, b) := ext-gcd(y, mod(x, y))
    return (d, b, a - floor(x/y) * b)
```

Recursively: $d = ay + b(x - \lfloor \frac{x}{y} \rfloor \cdot y)$

Review Proof: step.

```
ext-gcd(x, y)
  if y = 0 then return(x, 1, 0)
  else
    (d, a, b) := ext-gcd(y, mod(x, y))
    return (d, b, a - floor(x/y) * b)
```

Recursively: $d = ay + b(x - \lfloor \frac{x}{y} \rfloor \cdot y) \implies d = bx - (a - \lfloor \frac{x}{y} \rfloor b)y$

Review Proof: step.

```
ext-gcd(x, y)
  if y = 0 then return(x, 1, 0)
  else
    (d, a, b) := ext-gcd(y, mod(x, y))
    return (d, b, a - floor(x/y) * b)
```

Recursively: $d = ay + b(x - \lfloor \frac{x}{y} \rfloor \cdot y) \implies d = bx - (a - \lfloor \frac{x}{y} \rfloor b)y$

Returns $(d, b, (a - \lfloor \frac{x}{y} \rfloor \cdot b))$.

Hand Calculation Method for Inverses.

Example: $\gcd(7, 60) = 1$.

Hand Calculation Method for Inverses.

Example: $\gcd(7, 60) = 1$.
egcd(7,60).

Hand Calculation Method for Inverses.

Example: $\gcd(7, 60) = 1$.
egcd(7,60).

$$7(0) + 60(1) = 60$$

Hand Calculation Method for Inverses.

Example: $\gcd(7, 60) = 1$.
egcd(7,60).

$$7(0) + 60(1) = 60$$

$$7(1) + 60(0) = 7$$

Hand Calculation Method for Inverses.

Example: $\gcd(7, 60) = 1$.
egcd(7,60).

$$7(0) + 60(1) = 60$$

$$7(1) + 60(0) = 7$$

$$7(-8) + 60(1) = 4$$

Hand Calculation Method for Inverses.

Example: $\gcd(7, 60) = 1$.
egcd(7,60).

$$7(0) + 60(1) = 60$$

$$7(1) + 60(0) = 7$$

$$7(-8) + 60(1) = 4$$

$$7(9) + 60(-1) = 3$$

Hand Calculation Method for Inverses.

Example: $\gcd(7, 60) = 1$.
egcd(7,60).

$$7(0) + 60(1) = 60$$

$$7(1) + 60(0) = 7$$

$$7(-8) + 60(1) = 4$$

$$7(9) + 60(-1) = 3$$

$$7(-17) + 60(2) = 1$$

Hand Calculation Method for Inverses.

Example: $\gcd(7, 60) = 1$.
egcd(7,60).

$$7(0) + 60(1) = 60$$

$$7(1) + 60(0) = 7$$

$$7(-8) + 60(1) = 4$$

$$7(9) + 60(-1) = 3$$

$$7(-17) + 60(2) = 1$$

Hand Calculation Method for Inverses.

Example: $\gcd(7, 60) = 1$.
egcd(7,60).

$$\begin{aligned}7(0) + 60(1) &= 60 \\7(1) + 60(0) &= 7 \\7(-8) + 60(1) &= 4 \\7(9) + 60(-1) &= 3 \\7(-17) + 60(2) &= 1\end{aligned}$$

Confirm:

Hand Calculation Method for Inverses.

Example: $\gcd(7, 60) = 1$.
egcd(7,60).

$$\begin{aligned}7(0) + 60(1) &= 60 \\7(1) + 60(0) &= 7 \\7(-8) + 60(1) &= 4 \\7(9) + 60(-1) &= 3 \\7(-17) + 60(2) &= 1\end{aligned}$$

Confirm: $-119 + 120 = 1$

Wrap-up

Conclusion: Can find multiplicative inverses in $O(n)$ time!

Wrap-up

Conclusion: Can find multiplicative inverses in $O(n)$ time!

Very different from elementary school: try 1, try 2, try 3...

Wrap-up

Conclusion: Can find multiplicative inverses in $O(n)$ time!

Very different from elementary school: try 1, try 2, try 3...

$$2^{n/2}$$

Wrap-up

Conclusion: Can find multiplicative inverses in $O(n)$ time!

Very different from elementary school: try 1, try 2, try 3...

$$2^{n/2}$$

Inverse of 500,000,357 modulo 1,000,000,000,000?

Wrap-up

Conclusion: Can find multiplicative inverses in $O(n)$ time!

Very different from elementary school: try 1, try 2, try 3...

$$2^{n/2}$$

Inverse of 500,000,357 modulo 1,000,000,000,000?

≤ 80 divisions.

Wrap-up

Conclusion: Can find multiplicative inverses in $O(n)$ time!

Very different from elementary school: try 1, try 2, try 3...

$$2^{n/2}$$

Inverse of 500,000,357 modulo 1,000,000,000,000?

≤ 80 divisions.

versus 1,000,000

Wrap-up

Conclusion: Can find multiplicative inverses in $O(n)$ time!

Very different from elementary school: try 1, try 2, try 3...

$$2^{n/2}$$

Inverse of 500,000,357 modulo 1,000,000,000,000?

≤ 80 divisions.

versus 1,000,000

Wrap-up

Conclusion: Can find multiplicative inverses in $O(n)$ time!

Very different from elementary school: try 1, try 2, try 3...

$$2^{n/2}$$

Inverse of 500,000,357 modulo 1,000,000,000,000?

≤ 80 divisions.

versus 1,000,000

Internet Security.

Wrap-up

Conclusion: Can find multiplicative inverses in $O(n)$ time!

Very different from elementary school: try 1, try 2, try 3...

$$2^{n/2}$$

Inverse of 500,000,357 modulo 1,000,000,000,000?

≤ 80 divisions.

versus 1,000,000

Internet Security.

Public Key Cryptography: 512 digits.

Wrap-up

Conclusion: Can find multiplicative inverses in $O(n)$ time!

Very different from elementary school: try 1, try 2, try 3...

$$2^{n/2}$$

Inverse of 500,000,357 modulo 1,000,000,000,000?

≤ 80 divisions.

versus 1,000,000

Internet Security.

Public Key Cryptography: 512 digits.

512 divisions vs.

More Number Theory Tools.

Chinese remainder theorem.

More Number Theory Tools.

Chinese remainder theorem.

Fermat's Theorem.

Bijections

Bijection is **one to one** and **onto**.

Bijection:

Bijections

Bijection is **one to one** and **onto**.

Bijection:

$$f : A \rightarrow B.$$

Bijections

Bijection is **one to one** and **onto**.

Bijection:

$$f : A \rightarrow B.$$

Domain: A , Co-Domain: B .

Bijections

Bijection is **one to one** and **onto**.

Bijection:

$$f : A \rightarrow B.$$

Domain: A , Co-Domain: B .

Versus Range.

Bijections

Bijection is **one to one** and **onto**.

Bijection:

$$f : A \rightarrow B.$$

Domain: A , Co-Domain: B .

Versus Range.

E.g. **sin** (x).

Bijections

Bijection is **one to one** and **onto**.

Bijection:

$$f : A \rightarrow B.$$

Domain: A , Co-Domain: B .

Versus Range.

E.g. **sin** (x).

$$A = B = \text{reals.}$$

Bijections

Bijection is **one to one** and **onto**.

Bijection:

$$f : A \rightarrow B.$$

Domain: A , Co-Domain: B .

Versus Range.

E.g. **sin** (x).

$$A = B = \text{reals.}$$

Range is $[-1, 1]$.

Bijections

Bijection is **one to one** and **onto**.

Bijection:

$$f : A \rightarrow B.$$

Domain: A , Co-Domain: B .

Versus Range.

E.g. **sin** (x).

$$A = B = \text{reals.}$$

Range is $[-1, 1]$. Onto: $[-1, 1]$.

Bijections

Bijection is **one to one** and **onto**.

Bijection:

$$f : A \rightarrow B.$$

Domain: A , Co-Domain: B .

Versus Range.

E.g. **sin** (x).

$$A = B = \text{reals.}$$

Range is $[-1, 1]$. Onto: $[-1, 1]$.

Not one-to-one.

Bijections

Bijection is **one to one** and **onto**.

Bijection:

$$f : A \rightarrow B.$$

Domain: A , Co-Domain: B .

Versus Range.

E.g. **sin** (x).

$$A = B = \text{reals.}$$

Range is $[-1, 1]$. Onto: $[-1, 1]$.

Not one-to-one. **sin** (π) = **sin** (0) = 0.

Bijections

Bijection is **one to one** and **onto**.

Bijection:

$$f : A \rightarrow B.$$

Domain: A , Co-Domain: B .

Versus Range.

E.g. **sin** (x).

$$A = B = \text{reals.}$$

Range is $[-1, 1]$. Onto: $[-1, 1]$.

Not one-to-one. **sin** (π) = **sin** (0) = 0.

Range Definition always is onto.

Bijections

Bijection is **one to one** and **onto**.

Bijection:

$$f : A \rightarrow B.$$

Domain: A , Co-Domain: B .

Versus Range.

E.g. **sin** (x).

$$A = B = \text{reals.}$$

Range is $[-1, 1]$. Onto: $[-1, 1]$.

Not one-to-one. **sin** (π) = **sin** (0) = 0.

Range Definition always is onto.

Consider $f(x) = ax \pmod{m}$.

Bijections

Bijection is **one to one** and **onto**.

Bijection:

$$f : A \rightarrow B.$$

Domain: A , Co-Domain: B .

Versus Range.

E.g. **sin** (x).

$$A = B = \text{reals.}$$

Range is $[-1, 1]$. Onto: $[-1, 1]$.

Not one-to-one. **sin** (π) = **sin** (0) = 0.

Range Definition always is onto.

Consider $f(x) = ax \pmod{m}$.

$$f : \{0, \dots, m-1\} \rightarrow \{0, \dots, m-1\}.$$

Bijections

Bijection is **one to one** and **onto**.

Bijection:

$$f : A \rightarrow B.$$

Domain: A , Co-Domain: B .

Versus Range.

E.g. **sin** (x).

$$A = B = \text{reals.}$$

Range is $[-1, 1]$. Onto: $[-1, 1]$.

Not one-to-one. **sin** (π) = **sin** (0) = 0.

Range Definition always is onto.

Consider $f(x) = ax \pmod m$.

$$f : \{0, \dots, m-1\} \rightarrow \{0, \dots, m-1\}.$$

Domain/Co-Domain: $\{0, \dots, m-1\}$.

Bijections

Bijection is **one to one** and **onto**.

Bijection:

$$f : A \rightarrow B.$$

Domain: A , Co-Domain: B .

Versus Range.

E.g. **sin** (x).

$$A = B = \text{reals.}$$

Range is $[-1, 1]$. Onto: $[-1, 1]$.

Not one-to-one. **sin** (π) = **sin** (0) = 0.

Range Definition always is onto.

Consider $f(x) = ax \pmod{m}$.

$$f : \{0, \dots, m-1\} \rightarrow \{0, \dots, m-1\}.$$

Domain/Co-Domain: $\{0, \dots, m-1\}$.

When is it a bijection?

Bijections

Bijection is **one to one** and **onto**.

Bijection:

$$f : A \rightarrow B.$$

Domain: A , Co-Domain: B .

Versus Range.

E.g. **sin** (x).

$$A = B = \text{reals.}$$

Range is $[-1, 1]$. Onto: $[-1, 1]$.

Not one-to-one. **sin** (π) = **sin** (0) = 0.

Range Definition always is onto.

Consider $f(x) = ax \pmod{m}$.

$$f : \{0, \dots, m-1\} \rightarrow \{0, \dots, m-1\}.$$

Domain/Co-Domain: $\{0, \dots, m-1\}$.

When is it a bijection?

When $\gcd(a, m)$ is

Bijections

Bijection is **one to one** and **onto**.

Bijection:

$$f : A \rightarrow B.$$

Domain: A , Co-Domain: B .

Versus Range.

E.g. **sin** (x).

$$A = B = \text{reals.}$$

Range is $[-1, 1]$. Onto: $[-1, 1]$.

Not one-to-one. **sin** (π) = **sin** (0) = 0.

Range Definition always is onto.

Consider $f(x) = ax \pmod{m}$.

$$f : \{0, \dots, m-1\} \rightarrow \{0, \dots, m-1\}.$$

Domain/Co-Domain: $\{0, \dots, m-1\}$.

When is it a bijection?

When $\gcd(a, m)$ is?

Bijections

Bijection is **one to one** and **onto**.

Bijection:

$$f : A \rightarrow B.$$

Domain: A , Co-Domain: B .

Versus Range.

E.g. **sin** (x).

$$A = B = \text{reals.}$$

Range is $[-1, 1]$. Onto: $[-1, 1]$.

Not one-to-one. **sin** (π) = **sin** (0) = 0.

Range Definition always is onto.

Consider $f(x) = ax \pmod m$.

$$f : \{0, \dots, m-1\} \rightarrow \{0, \dots, m-1\}.$$

Domain/Co-Domain: $\{0, \dots, m-1\}$.

When is it a bijection?

When $\gcd(a, m)$ is? ... 1.

Bijections

Bijection is **one to one** and **onto**.

Bijection:

$$f : A \rightarrow B.$$

Domain: A , Co-Domain: B .

Versus Range.

E.g. **sin** (x).

$$A = B = \text{reals.}$$

Range is $[-1, 1]$. Onto: $[-1, 1]$.

Not one-to-one. **sin** (π) = **sin** (0) = 0.

Range Definition always is onto.

Consider $f(x) = ax \pmod{m}$.

$$f : \{0, \dots, m-1\} \rightarrow \{0, \dots, m-1\}.$$

Domain/Co-Domain: $\{0, \dots, m-1\}$.

When is it a bijection?

When $\gcd(a, m)$ is? ... 1.

Not Example: $a = 2, m = 4,$

Bijections

Bijection is **one to one** and **onto**.

Bijection:

$$f : A \rightarrow B.$$

Domain: A , Co-Domain: B .

Versus Range.

E.g. **sin** (x).

$$A = B = \text{reals.}$$

Range is $[-1, 1]$. Onto: $[-1, 1]$.

Not one-to-one. **sin** (π) = **sin** (0) = 0.

Range Definition always is onto.

Consider $f(x) = ax \pmod{m}$.

$$f : \{0, \dots, m-1\} \rightarrow \{0, \dots, m-1\}.$$

Domain/Co-Domain: $\{0, \dots, m-1\}$.

When is it a bijection?

When $\gcd(a, m)$ is? ... 1.

Not Example: $a = 2, m = 4, f(0) = f(2) = 0 \pmod{4}$.

Lots of Mods

$$x = 5 \pmod{7} \text{ and } x = 3 \pmod{5}.$$

Lots of Mods

$x = 5 \pmod{7}$ and $x = 3 \pmod{5}$.

What is $x \pmod{35}$?

Lots of Mods

$x = 5 \pmod{7}$ and $x = 3 \pmod{5}$.

What is $x \pmod{35}$?

Let's try 5.

Lots of Mods

$x = 5 \pmod{7}$ and $x = 3 \pmod{5}$.

What is $x \pmod{35}$?

Let's try 5. Not $3 \pmod{5}$!

Lots of Mods

$x = 5 \pmod{7}$ and $x = 3 \pmod{5}$.

What is $x \pmod{35}$?

Let's try 5. Not $3 \pmod{5}$!

Let's try 3.

Lots of Mods

$x = 5 \pmod{7}$ and $x = 3 \pmod{5}$.

What is $x \pmod{35}$?

Let's try 5. Not 3 (mod 5)!

Let's try 3. Not 5 (mod 7)!

Lots of Mods

$x = 5 \pmod{7}$ and $x = 3 \pmod{5}$.

What is $x \pmod{35}$?

Let's try 5. Not 3 $\pmod{5}$!

Let's try 3. Not 5 $\pmod{7}$!

Lots of Mods

$x = 5 \pmod{7}$ and $x = 3 \pmod{5}$.

What is $x \pmod{35}$?

Let's try 5. Not 3 $\pmod{5}$!

Let's try 3. Not 5 $\pmod{7}$!

If $x = 5 \pmod{7}$

Lots of Mods

$x = 5 \pmod{7}$ and $x = 3 \pmod{5}$.

What is $x \pmod{35}$?

Let's try 5. Not 3 $\pmod{5}$!

Let's try 3. Not 5 $\pmod{7}$!

If $x = 5 \pmod{7}$

then x is in $\{5, 12, 19, 26, 33\}$.

Lots of Mods

$x = 5 \pmod{7}$ and $x = 3 \pmod{5}$.

What is $x \pmod{35}$?

Let's try 5. Not 3 $\pmod{5}$!

Let's try 3. Not 5 $\pmod{7}$!

If $x = 5 \pmod{7}$

then x is in $\{5, 12, 19, 26, 33\}$.

Lots of Mods

$x = 5 \pmod{7}$ and $x = 3 \pmod{5}$.

What is $x \pmod{35}$?

Let's try 5. Not $3 \pmod{5}$!

Let's try 3. Not $5 \pmod{7}$!

If $x = 5 \pmod{7}$

then x is in $\{5, 12, 19, 26, 33\}$.

Oh, only 33 is $3 \pmod{5}$.

Lots of Mods

$x = 5 \pmod{7}$ and $x = 3 \pmod{5}$.

What is $x \pmod{35}$?

Let's try 5. Not $3 \pmod{5}$!

Let's try 3. Not $5 \pmod{7}$!

If $x = 5 \pmod{7}$

then x is in $\{5, 12, 19, 26, 33\}$.

Oh, only 33 is $3 \pmod{5}$.

Hmmm...

Lots of Mods

$x = 5 \pmod{7}$ and $x = 3 \pmod{5}$.

What is $x \pmod{35}$?

Let's try 5. Not $3 \pmod{5}$!

Let's try 3. Not $5 \pmod{7}$!

If $x = 5 \pmod{7}$

then x is in $\{5, 12, 19, 26, 33\}$.

Oh, only 33 is $3 \pmod{5}$.

Hmmm... only one solution.

Lots of Mods

$x = 5 \pmod{7}$ and $x = 3 \pmod{5}$.

What is $x \pmod{35}$?

Let's try 5. Not $3 \pmod{5}$!

Let's try 3. Not $5 \pmod{7}$!

If $x = 5 \pmod{7}$

then x is in $\{5, 12, 19, 26, 33\}$.

Oh, only 33 is $3 \pmod{5}$.

Hmmm... only one solution.

A bit slow for large values.

Simple Chinese Remainder Theorem.

Simple Chinese Remainder Theorem.

My love is won.

Simple Chinese Remainder Theorem.

My love is won. Zero and One.

Simple Chinese Remainder Theorem.

My love is won. Zero and One. Nothing and nothing done.

Simple Chinese Remainder Theorem.

My love is won. Zero and One. Nothing and nothing done.

Simple Chinese Remainder Theorem.

My love is won. Zero and One. Nothing and nothing done.

Find $x = a \pmod{m}$ and $x = b \pmod{n}$

Simple Chinese Remainder Theorem.

My love is won. Zero and One. Nothing and nothing done.

Find $x = a \pmod{m}$ and $x = b \pmod{n}$ where $\gcd(m, n) = 1$.

Simple Chinese Remainder Theorem.

My love is won. Zero and One. Nothing and nothing done.

Find $x = a \pmod{m}$ and $x = b \pmod{n}$ where $\gcd(m, n) = 1$.

CRT Thm: There is a unique solution $x \pmod{mn}$.

Simple Chinese Remainder Theorem.

My love is won. Zero and One. Nothing and nothing done.

Find $x = a \pmod{m}$ and $x = b \pmod{n}$ where $\gcd(m, n) = 1$.

CRT Thm: There is a unique solution $x \pmod{mn}$.

Proof:

Simple Chinese Remainder Theorem.

My love is won. Zero and One. Nothing and nothing done.

Find $x = a \pmod{m}$ and $x = b \pmod{n}$ where $\gcd(m, n) = 1$.

CRT Thm: There is a unique solution $x \pmod{mn}$.

Proof:

Consider $u = n(n^{-1} \pmod{m})$.

Simple Chinese Remainder Theorem.

My love is won. Zero and One. Nothing and nothing done.

Find $x = a \pmod{m}$ and $x = b \pmod{n}$ where $\gcd(m, n) = 1$.

CRT Thm: There is a unique solution $x \pmod{mn}$.

Proof:

Consider $u = n(n^{-1} \pmod{m})$.

$$u = 0 \pmod{n}$$

Simple Chinese Remainder Theorem.

My love is won. Zero and One. Nothing and nothing done.

Find $x = a \pmod{m}$ and $x = b \pmod{n}$ where $\gcd(m, n) = 1$.

CRT Thm: There is a unique solution $x \pmod{mn}$.

Proof:

Consider $u = n(n^{-1} \pmod{m})$.

$$u = 0 \pmod{n} \qquad u = 1 \pmod{m}$$

Simple Chinese Remainder Theorem.

My love is won. Zero and One. Nothing and nothing done.

Find $x = a \pmod{m}$ and $x = b \pmod{n}$ where $\gcd(m, n) = 1$.

CRT Thm: There is a unique solution $x \pmod{mn}$.

Proof:

Consider $u = n(n^{-1} \pmod{m})$.

$$u = 0 \pmod{n} \quad u = 1 \pmod{m}$$

Consider $v = m(m^{-1} \pmod{n})$.

Simple Chinese Remainder Theorem.

My love is won. Zero and One. Nothing and nothing done.

Find $x = a \pmod{m}$ and $x = b \pmod{n}$ where $\gcd(m, n) = 1$.

CRT Thm: There is a unique solution $x \pmod{mn}$.

Proof:

Consider $u = n(n^{-1} \pmod{m})$.

$$u = 0 \pmod{n} \quad u = 1 \pmod{m}$$

Consider $v = m(m^{-1} \pmod{n})$.

$$v = 1 \pmod{n}$$

Simple Chinese Remainder Theorem.

My love is won. Zero and One. Nothing and nothing done.

Find $x = a \pmod{m}$ and $x = b \pmod{n}$ where $\gcd(m, n) = 1$.

CRT Thm: There is a unique solution $x \pmod{mn}$.

Proof:

Consider $u = n(n^{-1} \pmod{m})$.

$$u = 0 \pmod{n} \quad u = 1 \pmod{m}$$

Consider $v = m(m^{-1} \pmod{n})$.

$$v = 1 \pmod{n} \quad v = 0 \pmod{m}$$

Simple Chinese Remainder Theorem.

My love is won. Zero and One. Nothing and nothing done.

Find $x = a \pmod{m}$ and $x = b \pmod{n}$ where $\gcd(m, n) = 1$.

CRT Thm: There is a unique solution $x \pmod{mn}$.

Proof:

Consider $u = n(n^{-1} \pmod{m})$.

$$u = 0 \pmod{n} \quad u = 1 \pmod{m}$$

Consider $v = m(m^{-1} \pmod{n})$.

$$v = 1 \pmod{n} \quad v = 0 \pmod{m}$$

Simple Chinese Remainder Theorem.

My love is won. Zero and One. Nothing and nothing done.

Find $x = a \pmod{m}$ and $x = b \pmod{n}$ where $\gcd(m, n) = 1$.

CRT Thm: There is a unique solution $x \pmod{mn}$.

Proof:

Consider $u = n(n^{-1} \pmod{m})$.

$$u = 0 \pmod{n} \quad u = 1 \pmod{m}$$

Consider $v = m(m^{-1} \pmod{n})$.

$$v = 1 \pmod{n} \quad v = 0 \pmod{m}$$

Let $x = au + bv$.

Simple Chinese Remainder Theorem.

My love is won. Zero and One. Nothing and nothing done.

Find $x = a \pmod{m}$ and $x = b \pmod{n}$ where $\gcd(m, n) = 1$.

CRT Thm: There is a unique solution $x \pmod{mn}$.

Proof:

Consider $u = n(n^{-1} \pmod{m})$.

$$u = 0 \pmod{n} \quad u = 1 \pmod{m}$$

Consider $v = m(m^{-1} \pmod{n})$.

$$v = 1 \pmod{n} \quad v = 0 \pmod{m}$$

Let $x = au + bv$.

$$x = au + bv = a \pmod{m}$$

Simple Chinese Remainder Theorem.

My love is won. Zero and One. Nothing and nothing done.

Find $x = a \pmod{m}$ and $x = b \pmod{n}$ where $\gcd(m, n) = 1$.

CRT Thm: There is a unique solution $x \pmod{mn}$.

Proof:

Consider $u = n(n^{-1} \pmod{m})$.

$$u = 0 \pmod{n} \quad u = 1 \pmod{m}$$

Consider $v = m(m^{-1} \pmod{n})$.

$$v = 1 \pmod{n} \quad v = 0 \pmod{m}$$

Let $x = au + bv$.

$$x = au + bv = a \pmod{m} : bv = 0 \pmod{m} \text{ and } au = a \pmod{m}$$

Simple Chinese Remainder Theorem.

My love is won. Zero and One. Nothing and nothing done.

Find $x = a \pmod{m}$ and $x = b \pmod{n}$ where $\gcd(m, n) = 1$.

CRT Thm: There is a unique solution $x \pmod{mn}$.

Proof:

Consider $u = n(n^{-1} \pmod{m})$.

$$u = 0 \pmod{n} \quad u = 1 \pmod{m}$$

Consider $v = m(m^{-1} \pmod{n})$.

$$v = 1 \pmod{n} \quad v = 0 \pmod{m}$$

Let $x = au + bv$.

$$x = au + bv = a \pmod{m} : bv = 0 \pmod{m} \text{ and } au = a \pmod{m}$$

Simple Chinese Remainder Theorem.

My love is won. Zero and One. Nothing and nothing done.

Find $x = a \pmod{m}$ and $x = b \pmod{n}$ where $\gcd(m, n) = 1$.

CRT Thm: There is a unique solution $x \pmod{mn}$.

Proof:

Consider $u = n(n^{-1} \pmod{m})$.

$$u = 0 \pmod{n} \quad u = 1 \pmod{m}$$

Consider $v = m(m^{-1} \pmod{n})$.

$$v = 1 \pmod{n} \quad v = 0 \pmod{m}$$

Let $x = au + bv$.

$$x = au + bv = a \pmod{m} : bv = 0 \pmod{m} \text{ and } au = a \pmod{m}$$

$$x = au + bv = b \pmod{n}$$

Simple Chinese Remainder Theorem.

My love is won. Zero and One. Nothing and nothing done.

Find $x = a \pmod{m}$ and $x = b \pmod{n}$ where $\gcd(m, n) = 1$.

CRT Thm: There is a unique solution $x \pmod{mn}$.

Proof:

Consider $u = n(n^{-1} \pmod{m})$.

$$u = 0 \pmod{n} \quad u = 1 \pmod{m}$$

Consider $v = m(m^{-1} \pmod{n})$.

$$v = 1 \pmod{n} \quad v = 0 \pmod{m}$$

Let $x = au + bv$.

$$x = au + bv = a \pmod{m} : bv = 0 \pmod{m} \text{ and } au = a \pmod{m}$$

$$x = au + bv = b \pmod{n} : au = 0 \pmod{n} \text{ and } bv = b \pmod{n}$$

Simple Chinese Remainder Theorem.

My love is won. Zero and One. Nothing and nothing done.

Find $x = a \pmod{m}$ and $x = b \pmod{n}$ where $\gcd(m, n) = 1$.

CRT Thm: There is a unique solution $x \pmod{mn}$.

Proof:

Consider $u = n(n^{-1} \pmod{m})$.

$$u = 0 \pmod{n} \quad u = 1 \pmod{m}$$

Consider $v = m(m^{-1} \pmod{n})$.

$$v = 1 \pmod{n} \quad v = 0 \pmod{m}$$

Let $x = au + bv$.

$$x = au + bv = a \pmod{m} : bv = 0 \pmod{m} \text{ and } au = a \pmod{m}$$

$$x = au + bv = b \pmod{n} : au = 0 \pmod{n} \text{ and } bv = b \pmod{n}$$

Simple Chinese Remainder Theorem.

My love is won. Zero and One. Nothing and nothing done.

Find $x = a \pmod{m}$ and $x = b \pmod{n}$ where $\gcd(m, n) = 1$.

CRT Thm: There is a unique solution $x \pmod{mn}$.

Proof:

Consider $u = n(n^{-1} \pmod{m})$.

$$u = 0 \pmod{n} \quad u = 1 \pmod{m}$$

Consider $v = m(m^{-1} \pmod{n})$.

$$v = 1 \pmod{n} \quad v = 0 \pmod{m}$$

Let $x = au + bv$.

$$x = au + bv = a \pmod{m} : bv = 0 \pmod{m} \text{ and } au = a \pmod{m}$$

$$x = au + bv = b \pmod{n} : au = 0 \pmod{n} \text{ and } bv = b \pmod{n}$$

Only solution?

Simple Chinese Remainder Theorem.

My love is won. Zero and One. Nothing and nothing done.

Find $x = a \pmod{m}$ and $x = b \pmod{n}$ where $\gcd(m, n) = 1$.

CRT Thm: There is a unique solution $x \pmod{mn}$.

Proof:

Consider $u = n(n^{-1} \pmod{m})$.

$$u = 0 \pmod{n} \quad u = 1 \pmod{m}$$

Consider $v = m(m^{-1} \pmod{n})$.

$$v = 1 \pmod{n} \quad v = 0 \pmod{m}$$

Let $x = au + bv$.

$$x = au + bv = a \pmod{m} : bv = 0 \pmod{m} \text{ and } au = a \pmod{m}$$

$$x = au + bv = b \pmod{n} : au = 0 \pmod{n} \text{ and } bv = b \pmod{n}$$

Only solution? If not, two solutions, x and y .

Simple Chinese Remainder Theorem.

My love is won. Zero and One. Nothing and nothing done.

Find $x = a \pmod{m}$ and $x = b \pmod{n}$ where $\gcd(m, n) = 1$.

CRT Thm: There is a unique solution $x \pmod{mn}$.

Proof:

Consider $u = n(n^{-1} \pmod{m})$.

$$u = 0 \pmod{n} \quad u = 1 \pmod{m}$$

Consider $v = m(m^{-1} \pmod{n})$.

$$v = 1 \pmod{n} \quad v = 0 \pmod{m}$$

Let $x = au + bv$.

$$x = au + bv = a \pmod{m} : bv = 0 \pmod{m} \text{ and } au = a \pmod{m}$$

$$x = au + bv = b \pmod{n} : au = 0 \pmod{n} \text{ and } bv = b \pmod{n}$$

Only solution? If not, two solutions, x and y .

$$(x - y) \equiv 0 \pmod{m} \text{ and } (x - y) \equiv 0 \pmod{n}.$$

Simple Chinese Remainder Theorem.

My love is won. Zero and One. Nothing and nothing done.

Find $x = a \pmod{m}$ and $x = b \pmod{n}$ where $\gcd(m, n) = 1$.

CRT Thm: There is a unique solution $x \pmod{mn}$.

Proof:

Consider $u = n(n^{-1} \pmod{m})$.

$$u = 0 \pmod{n} \quad u = 1 \pmod{m}$$

Consider $v = m(m^{-1} \pmod{n})$.

$$v = 1 \pmod{n} \quad v = 0 \pmod{m}$$

Let $x = au + bv$.

$$x = au + bv = a \pmod{m} : bv = 0 \pmod{m} \text{ and } au = a \pmod{m}$$

$$x = au + bv = b \pmod{n} : au = 0 \pmod{n} \text{ and } bv = b \pmod{n}$$

Only solution? If not, two solutions, x and y .

$$(x - y) \equiv 0 \pmod{m} \text{ and } (x - y) \equiv 0 \pmod{n}.$$

$\implies (x - y)$ is multiple of m and n since $\gcd(m, n) = 1$.

Simple Chinese Remainder Theorem.

My love is won. Zero and One. Nothing and nothing done.

Find $x = a \pmod{m}$ and $x = b \pmod{n}$ where $\gcd(m, n) = 1$.

CRT Thm: There is a unique solution $x \pmod{mn}$.

Proof:

Consider $u = n(n^{-1} \pmod{m})$.

$$u = 0 \pmod{n} \quad u = 1 \pmod{m}$$

Consider $v = m(m^{-1} \pmod{n})$.

$$v = 1 \pmod{n} \quad v = 0 \pmod{m}$$

Let $x = au + bv$.

$$x = au + bv = a \pmod{m} : bv = 0 \pmod{m} \text{ and } au = a \pmod{m}$$

$$x = au + bv = b \pmod{n} : au = 0 \pmod{n} \text{ and } bv = b \pmod{n}$$

Only solution? If not, two solutions, x and y .

$$(x - y) \equiv 0 \pmod{m} \text{ and } (x - y) \equiv 0 \pmod{n}.$$

$$\implies (x - y) \text{ is multiple of } m \text{ and } n \text{ since } \gcd(m, n) = 1.$$

$$\implies x - y \geq mn$$

Simple Chinese Remainder Theorem.

My love is won. Zero and One. Nothing and nothing done.

Find $x = a \pmod{m}$ and $x = b \pmod{n}$ where $\gcd(m, n) = 1$.

CRT Thm: There is a unique solution $x \pmod{mn}$.

Proof:

Consider $u = n(n^{-1} \pmod{m})$.

$$u = 0 \pmod{n} \quad u = 1 \pmod{m}$$

Consider $v = m(m^{-1} \pmod{n})$.

$$v = 1 \pmod{n} \quad v = 0 \pmod{m}$$

Let $x = au + bv$.

$$x = au + bv = a \pmod{m} : bv = 0 \pmod{m} \text{ and } au = a \pmod{m}$$

$$x = au + bv = b \pmod{n} : au = 0 \pmod{n} \text{ and } bv = b \pmod{n}$$

Only solution? If not, two solutions, x and y .

$$(x - y) \equiv 0 \pmod{m} \text{ and } (x - y) \equiv 0 \pmod{n}.$$

$$\implies (x - y) \text{ is multiple of } m \text{ and } n \text{ since } \gcd(m, n) = 1.$$

$$\implies x - y \geq mn \implies x, y \notin \{0, \dots, mn - 1\}.$$

Simple Chinese Remainder Theorem.

My love is won. Zero and One. Nothing and nothing done.

Find $x = a \pmod{m}$ and $x = b \pmod{n}$ where $\gcd(m, n) = 1$.

CRT Thm: There is a unique solution $x \pmod{mn}$.

Proof:

Consider $u = n(n^{-1} \pmod{m})$.

$$u = 0 \pmod{n} \quad u = 1 \pmod{m}$$

Consider $v = m(m^{-1} \pmod{n})$.

$$v = 1 \pmod{n} \quad v = 0 \pmod{m}$$

Let $x = au + bv$.

$$x = au + bv = a \pmod{m} : bv = 0 \pmod{m} \text{ and } au = a \pmod{m}$$

$$x = au + bv = b \pmod{n} : au = 0 \pmod{n} \text{ and } bv = b \pmod{n}$$

Only solution? If not, two solutions, x and y .

$$(x - y) \equiv 0 \pmod{m} \text{ and } (x - y) \equiv 0 \pmod{n}.$$

$$\implies (x - y) \text{ is multiple of } m \text{ and } n \text{ since } \gcd(m, n) = 1.$$

$$\implies x - y \geq mn \implies x, y \notin \{0, \dots, mn - 1\}.$$

Thus, only one solution modulo mn .

Simple Chinese Remainder Theorem.

My love is won. Zero and One. Nothing and nothing done.

Find $x = a \pmod{m}$ and $x = b \pmod{n}$ where $\gcd(m, n) = 1$.

CRT Thm: There is a unique solution $x \pmod{mn}$.

Proof:

Consider $u = n(n^{-1} \pmod{m})$.

$$u = 0 \pmod{n} \quad u = 1 \pmod{m}$$

Consider $v = m(m^{-1} \pmod{n})$.

$$v = 1 \pmod{n} \quad v = 0 \pmod{m}$$

Let $x = au + bv$.

$$x = au + bv = a \pmod{m} : bv = 0 \pmod{m} \text{ and } au = a \pmod{m}$$

$$x = au + bv = b \pmod{n} : au = 0 \pmod{n} \text{ and } bv = b \pmod{n}$$

Only solution? If not, two solutions, x and y .

$$(x - y) \equiv 0 \pmod{m} \text{ and } (x - y) \equiv 0 \pmod{n}.$$

$$\implies (x - y) \text{ is multiple of } m \text{ and } n \text{ since } \gcd(m, n) = 1.$$

$$\implies x - y \geq mn \implies x, y \notin \{0, \dots, mn - 1\}.$$

Thus, only one solution modulo mn .



Fermat's Theorem: Reducing Exponents.

Fermat's Little Theorem: For prime p , and $a \not\equiv 0 \pmod{p}$,

Fermat's Theorem: Reducing Exponents.

Fermat's Little Theorem: For prime p , and $a \not\equiv 0 \pmod{p}$,

$$a^{p-1} \equiv 1 \pmod{p}.$$

Fermat's Theorem: Reducing Exponents.

Fermat's Little Theorem: For prime p , and $a \not\equiv 0 \pmod{p}$,

$$a^{p-1} \equiv 1 \pmod{p}.$$

Proof:

Fermat's Theorem: Reducing Exponents.

Fermat's Little Theorem: For prime p , and $a \not\equiv 0 \pmod{p}$,

$$a^{p-1} \equiv 1 \pmod{p}.$$

Proof: Consider $S = \{a \cdot 1, \dots, a \cdot (p-1)\}$.

Fermat's Theorem: Reducing Exponents.

Fermat's Little Theorem: For prime p , and $a \not\equiv 0 \pmod{p}$,

$$a^{p-1} \equiv 1 \pmod{p}.$$

Proof: Consider $S = \{a \cdot 1, \dots, a \cdot (p-1)\}$.

All different modulo p since a has an inverse modulo p .

Fermat's Theorem: Reducing Exponents.

Fermat's Little Theorem: For prime p , and $a \not\equiv 0 \pmod{p}$,

$$a^{p-1} \equiv 1 \pmod{p}.$$

Proof: Consider $S = \{a \cdot 1, \dots, a \cdot (p-1)\}$.

All different modulo p since a has an inverse modulo p .

S contains representative of $\{1, \dots, p-1\}$ modulo p .

Fermat's Theorem: Reducing Exponents.

Fermat's Little Theorem: For prime p , and $a \not\equiv 0 \pmod{p}$,

$$a^{p-1} \equiv 1 \pmod{p}.$$

Proof: Consider $S = \{a \cdot 1, \dots, a \cdot (p-1)\}$.

All different modulo p since a has an inverse modulo p .

S contains representative of $\{1, \dots, p-1\}$ modulo p .

$$(a \cdot 1) \cdot (a \cdot 2) \cdots (a \cdot (p-1)) \equiv 1 \cdot 2 \cdots (p-1) \pmod{p},$$

Fermat's Theorem: Reducing Exponents.

Fermat's Little Theorem: For prime p , and $a \not\equiv 0 \pmod{p}$,

$$a^{p-1} \equiv 1 \pmod{p}.$$

Proof: Consider $S = \{a \cdot 1, \dots, a \cdot (p-1)\}$.

All different modulo p since a has an inverse modulo p .

S contains representative of $\{1, \dots, p-1\}$ modulo p .

$$(a \cdot 1) \cdot (a \cdot 2) \cdots (a \cdot (p-1)) \equiv 1 \cdot 2 \cdots (p-1) \pmod{p},$$

Since multiplication is commutative.

Fermat's Theorem: Reducing Exponents.

Fermat's Little Theorem: For prime p , and $a \not\equiv 0 \pmod{p}$,

$$a^{p-1} \equiv 1 \pmod{p}.$$

Proof: Consider $S = \{a \cdot 1, \dots, a \cdot (p-1)\}$.

All different modulo p since a has an inverse modulo p .

S contains representative of $\{1, \dots, p-1\}$ modulo p .

$$(a \cdot 1) \cdot (a \cdot 2) \cdots (a \cdot (p-1)) \equiv 1 \cdot 2 \cdots (p-1) \pmod{p},$$

Since multiplication is commutative.

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

Fermat's Theorem: Reducing Exponents.

Fermat's Little Theorem: For prime p , and $a \not\equiv 0 \pmod{p}$,

$$a^{p-1} \equiv 1 \pmod{p}.$$

Proof: Consider $S = \{a \cdot 1, \dots, a \cdot (p-1)\}$.

All different modulo p since a has an inverse modulo p .

S contains representative of $\{1, \dots, p-1\}$ modulo p .

$$(a \cdot 1) \cdot (a \cdot 2) \cdots (a \cdot (p-1)) \equiv 1 \cdot 2 \cdots (p-1) \pmod{p},$$

Since multiplication is commutative.

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

Each of $2, \dots, (p-1)$ has an inverse modulo p ,

Fermat's Theorem: Reducing Exponents.

Fermat's Little Theorem: For prime p , and $a \not\equiv 0 \pmod{p}$,

$$a^{p-1} \equiv 1 \pmod{p}.$$

Proof: Consider $S = \{a \cdot 1, \dots, a \cdot (p-1)\}$.

All different modulo p since a has an inverse modulo p .

S contains representative of $\{1, \dots, p-1\}$ modulo p .

$$(a \cdot 1) \cdot (a \cdot 2) \cdots (a \cdot (p-1)) \equiv 1 \cdot 2 \cdots (p-1) \pmod{p},$$

Since multiplication is commutative.

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

Each of $2, \dots, (p-1)$ has an inverse modulo p , solve to get...

Fermat's Theorem: Reducing Exponents.

Fermat's Little Theorem: For prime p , and $a \not\equiv 0 \pmod{p}$,

$$a^{p-1} \equiv 1 \pmod{p}.$$

Proof: Consider $S = \{a \cdot 1, \dots, a \cdot (p-1)\}$.

All different modulo p since a has an inverse modulo p .

S contains representative of $\{1, \dots, p-1\}$ modulo p .

$$(a \cdot 1) \cdot (a \cdot 2) \cdots (a \cdot (p-1)) \equiv 1 \cdot 2 \cdots (p-1) \pmod{p},$$

Since multiplication is commutative.

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

Each of $2, \dots, (p-1)$ has an inverse modulo p , solve to get...

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

Fermat's Theorem: Reducing Exponents.

Fermat's Little Theorem: For prime p , and $a \not\equiv 0 \pmod{p}$,

$$a^{p-1} \equiv 1 \pmod{p}.$$

Proof: Consider $S = \{a \cdot 1, \dots, a \cdot (p-1)\}$.

All different modulo p since a has an inverse modulo p .

S contains representative of $\{1, \dots, p-1\}$ modulo p .

$$(a \cdot 1) \cdot (a \cdot 2) \cdots (a \cdot (p-1)) \equiv 1 \cdot 2 \cdots (p-1) \pmod{p},$$

Since multiplication is commutative.

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

Each of $2, \dots, (p-1)$ has an inverse modulo p , solve to get...

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



Fermat and Exponent reducing.

Fermat's Little Theorem: For prime p , and $a \not\equiv 0 \pmod{p}$,

Fermat and Exponent reducing.

Fermat's Little Theorem: For prime p , and $a \not\equiv 0 \pmod{p}$,

$$a^{p-1} \equiv 1 \pmod{p}.$$

Fermat and Exponent reducing.

Fermat's Little Theorem: For prime p , and $a \not\equiv 0 \pmod{p}$,

$$a^{p-1} \equiv 1 \pmod{p}.$$

What is $2^{101} \pmod{7}$?

Fermat and Exponent reducing.

Fermat's Little Theorem: For prime p , and $a \not\equiv 0 \pmod{p}$,

$$a^{p-1} \equiv 1 \pmod{p}.$$

What is $2^{101} \pmod{7}$?

Wrong: $2^{101} = 2^{7*14+3} = 2^3 \pmod{7}$

Fermat and Exponent reducing.

Fermat's Little Theorem: For prime p , and $a \not\equiv 0 \pmod{p}$,

$$a^{p-1} \equiv 1 \pmod{p}.$$

What is $2^{101} \pmod{7}$?

Wrong: $2^{101} = 2^{7*14+3} = 2^3 \pmod{7}$

Fermat: 2 is relatively prime to 7. $\implies 2^6 = 1 \pmod{7}$.

Fermat and Exponent reducing.

Fermat's Little Theorem: For prime p , and $a \not\equiv 0 \pmod{p}$,

$$a^{p-1} \equiv 1 \pmod{p}.$$

What is $2^{101} \pmod{7}$?

Wrong: $2^{101} = 2^{7 \cdot 14 + 3} = 2^3 \pmod{7}$

Fermat: 2 is relatively prime to 7. $\implies 2^6 = 1 \pmod{7}$.

Correct: $2^{101} = 2^{6 \cdot 16 + 5} = 2^5 = 32 = 4 \pmod{7}$.

Fermat and Exponent reducing.

Fermat's Little Theorem: For prime p , and $a \not\equiv 0 \pmod{p}$,

$$a^{p-1} \equiv 1 \pmod{p}.$$

What is $2^{101} \pmod{7}$?

Wrong: $2^{101} = 2^{7 \cdot 14 + 3} = 2^3 \pmod{7}$

Fermat: 2 is relatively prime to 7. $\implies 2^6 = 1 \pmod{7}$.

Correct: $2^{101} = 2^{6 \cdot 16 + 5} = 2^5 = 32 = 4 \pmod{7}$.

For a prime modulus, we can reduce exponents modulo $p - 1$!

Isomorphisms.

Bijection:

Isomorphisms.

Bijection:

$$f(x) = ax \pmod{m} \text{ if } \gcd(a, m) = 1.$$

Isomorphisms.

Bijection:

$$f(x) = ax \pmod{m} \text{ if } \gcd(a, m) = 1.$$

Chinese Remainder Theorem: For m, n with $\gcd(n, m) = 1$,

Isomorphisms.

Bijection:

$$f(x) = ax \pmod{m} \text{ if } \gcd(a, m) = 1.$$

Chinese Remainder Theorem: For m, n with $\gcd(m, n) = 1$,
 \implies unique $x \pmod{mn}$ where $x = a \pmod{m}$ and $x = b \pmod{n}$.

Isomorphisms.

Bijection:

$$f(x) = ax \pmod{m} \text{ if } \gcd(a, m) = 1.$$

Chinese Remainder Theorem: For m, n with $\gcd(n, m) = 1$,
 \implies unique $x \pmod{mn}$ where $x = a \pmod{m}$ and $x = b \pmod{n}$.

Bijection between $(a \pmod{n}, b \pmod{m})$ and $x \pmod{mn}$.

Isomorphisms.

Bijection:

$$f(x) = ax \pmod{m} \text{ if } \gcd(a, m) = 1.$$

Chinese Remainder Theorem: For m, n with $\gcd(n, m) = 1$,
 \implies unique $x \pmod{mn}$ where $x = a \pmod{m}$ and $x = b \pmod{n}$.

Bijection between $(a \pmod{n}, b \pmod{m})$ and $x \pmod{mn}$.

Consider $m = 5, n = 9$, then if $(a, b) = (3, 7)$ then $x = 43 \pmod{45}$.

Isomorphisms.

Bijection:

$$f(x) = ax \pmod{m} \text{ if } \gcd(a, m) = 1.$$

Chinese Remainder Theorem: For m, n with $\gcd(n, m) = 1$,
 \implies unique $x \pmod{mn}$ where $x = a \pmod{m}$ and $x = b \pmod{n}$.

Bijection between $(a \pmod{n}, b \pmod{m})$ and $x \pmod{mn}$.

Consider $m = 5, n = 9$, then if $(a, b) = (3, 7)$ then $x = 43 \pmod{45}$.

Consider $(a', b') = (2, 4)$, then $x = 22 \pmod{45}$.

Isomorphisms.

Bijection:

$$f(x) = ax \pmod{m} \text{ if } \gcd(a, m) = 1.$$

Chinese Remainder Theorem: For m, n with $\gcd(n, m) = 1$,
 \implies unique $x \pmod{mn}$ where $x = a \pmod{m}$ and $x = b \pmod{n}$.

Bijection between $(a \pmod{n}, b \pmod{m})$ and $x \pmod{mn}$.

Consider $m = 5, n = 9$, then if $(a, b) = (3, 7)$ then $x = 43 \pmod{45}$.

Consider $(a', b') = (2, 4)$, then $x = 22 \pmod{45}$.

Now consider:

Isomorphisms.

Bijection:

$$f(x) = ax \pmod{m} \text{ if } \gcd(a, m) = 1.$$

Chinese Remainder Theorem: For m, n with $\gcd(n, m) = 1$,
 \implies unique $x \pmod{mn}$ where $x = a \pmod{m}$ and $x = b \pmod{n}$.

Bijection between $(a \pmod{n}, b \pmod{m})$ and $x \pmod{mn}$.

Consider $m = 5, n = 9$, then if $(a, b) = (3, 7)$ then $x = 43 \pmod{45}$.

Consider $(a', b') = (2, 4)$, then $x = 22 \pmod{45}$.

Now consider: $(a, b) + (a', b') = (0, 2)$.

Isomorphisms.

Bijection:

$$f(x) = ax \pmod{m} \text{ if } \gcd(a, m) = 1.$$

Chinese Remainder Theorem: For m, n with $\gcd(n, m) = 1$,
 \implies unique $x \pmod{mn}$ where $x = a \pmod{m}$ and $x = b \pmod{n}$.

Bijection between $(a \pmod{n}, b \pmod{m})$ and $x \pmod{mn}$.

Consider $m = 5, n = 9$, then if $(a, b) = (3, 7)$ then $x = 43 \pmod{45}$.

Consider $(a', b') = (2, 4)$, then $x = 22 \pmod{45}$.

Now consider: $(a, b) + (a', b') = (0, 2)$.

What is x where $x = 0 \pmod{5}$ and $x = 2 \pmod{9}$?

Isomorphisms.

Bijection:

$$f(x) = ax \pmod{m} \text{ if } \gcd(a, m) = 1.$$

Chinese Remainder Theorem: For m, n with $\gcd(n, m) = 1$,
 \implies unique $x \pmod{mn}$ where $x = a \pmod{m}$ and $x = b \pmod{n}$.

Bijection between $(a \pmod{n}, b \pmod{m})$ and $x \pmod{mn}$.

Consider $m = 5, n = 9$, then if $(a, b) = (3, 7)$ then $x = 43 \pmod{45}$.

Consider $(a', b') = (2, 4)$, then $x = 22 \pmod{45}$.

Now consider: $(a, b) + (a', b') = (0, 2)$.

What is x where $x = 0 \pmod{5}$ and $x = 2 \pmod{9}$?

Try $43 + 22 = 65$

Isomorphisms.

Bijection:

$$f(x) = ax \pmod{m} \text{ if } \gcd(a, m) = 1.$$

Chinese Remainder Theorem: For m, n with $\gcd(n, m) = 1$,
 \implies unique $x \pmod{mn}$ where $x = a \pmod{m}$ and $x = b \pmod{n}$.

Bijection between $(a \pmod{n}, b \pmod{m})$ and $x \pmod{mn}$.

Consider $m = 5, n = 9$, then if $(a, b) = (3, 7)$ then $x = 43 \pmod{45}$.

Consider $(a', b') = (2, 4)$, then $x = 22 \pmod{45}$.

Now consider: $(a, b) + (a', b') = (0, 2)$.

What is x where $x = 0 \pmod{5}$ and $x = 2 \pmod{9}$?

Try $43 + 22 = 65 = 20 \pmod{45}$.

Isomorphisms.

Bijection:

$$f(x) = ax \pmod{m} \text{ if } \gcd(a, m) = 1.$$

Chinese Remainder Theorem: For m, n with $\gcd(n, m) = 1$,
 \implies unique $x \pmod{mn}$ where $x = a \pmod{m}$ and $x = b \pmod{n}$.

Bijection between $(a \pmod{n}, b \pmod{m})$ and $x \pmod{mn}$.

Consider $m = 5, n = 9$, then if $(a, b) = (3, 7)$ then $x = 43 \pmod{45}$.

Consider $(a', b') = (2, 4)$, then $x = 22 \pmod{45}$.

Now consider: $(a, b) + (a', b') = (0, 2)$.

What is x where $x = 0 \pmod{5}$ and $x = 2 \pmod{9}$?

Try $43 + 22 = 65 = 20 \pmod{45}$.

Is it $0 \pmod{5}$?

Isomorphisms.

Bijection:

$$f(x) = ax \pmod{m} \text{ if } \gcd(a, m) = 1.$$

Chinese Remainder Theorem: For m, n with $\gcd(n, m) = 1$,
 \implies unique $x \pmod{mn}$ where $x = a \pmod{m}$ and $x = b \pmod{n}$.

Bijection between $(a \pmod{n}, b \pmod{m})$ and $x \pmod{mn}$.

Consider $m = 5, n = 9$, then if $(a, b) = (3, 7)$ then $x = 43 \pmod{45}$.

Consider $(a', b') = (2, 4)$, then $x = 22 \pmod{45}$.

Now consider: $(a, b) + (a', b') = (0, 2)$.

What is x where $x = 0 \pmod{5}$ and $x = 2 \pmod{9}$?

Try $43 + 22 = 65 = 20 \pmod{45}$.

Is it $0 \pmod{5}$? Yes!

Isomorphisms.

Bijection:

$f(x) = ax \pmod{m}$ if $\gcd(a, m) = 1$.

Chinese Remainder Theorem: For m, n with $\gcd(n, m) = 1$,
 \implies unique $x \pmod{mn}$ where $x = a \pmod{m}$ and $x = b \pmod{n}$.

Bijection between $(a \pmod{n}, b \pmod{m})$ and $x \pmod{mn}$.

Consider $m = 5, n = 9$, then if $(a, b) = (3, 7)$ then $x = 43 \pmod{45}$.

Consider $(a', b') = (2, 4)$, then $x = 22 \pmod{45}$.

Now consider: $(a, b) + (a', b') = (0, 2)$.

What is x where $x = 0 \pmod{5}$ and $x = 2 \pmod{9}$?

Try $43 + 22 = 65 = 20 \pmod{45}$.

Is it $0 \pmod{5}$? Yes! Is it $2 \pmod{9}$?

Isomorphisms.

Bijection:

$$f(x) = ax \pmod{m} \text{ if } \gcd(a, m) = 1.$$

Chinese Remainder Theorem: For m, n with $\gcd(n, m) = 1$,
 \implies unique $x \pmod{mn}$ where $x = a \pmod{m}$ and $x = b \pmod{n}$.

Bijection between $(a \pmod{n}, b \pmod{m})$ and $x \pmod{mn}$.

Consider $m = 5, n = 9$, then if $(a, b) = (3, 7)$ then $x = 43 \pmod{45}$.

Consider $(a', b') = (2, 4)$, then $x = 22 \pmod{45}$.

Now consider: $(a, b) + (a', b') = (0, 2)$.

What is x where $x = 0 \pmod{5}$ and $x = 2 \pmod{9}$?

Try $43 + 22 = 65 = 20 \pmod{45}$.

Is it $0 \pmod{5}$? Yes! Is it $2 \pmod{9}$? Yes!

Isomorphisms.

Bijection:

$$f(x) = ax \pmod{m} \text{ if } \gcd(a, m) = 1.$$

Chinese Remainder Theorem: For m, n with $\gcd(m, n) = 1$,
 \implies unique $x \pmod{mn}$ where $x = a \pmod{m}$ and $x = b \pmod{n}$.

Bijection between $(a \pmod{m}, b \pmod{n})$ and $x \pmod{mn}$.

Consider $m = 5, n = 9$, then if $(a, b) = (3, 7)$ then $x = 43 \pmod{45}$.

Consider $(a', b') = (2, 4)$, then $x = 22 \pmod{45}$.

Now consider: $(a, b) + (a', b') = (0, 2)$.

What is x where $x = 0 \pmod{5}$ and $x = 2 \pmod{9}$?

Try $43 + 22 = 65 = 20 \pmod{45}$.

Is it $0 \pmod{5}$? Yes! Is it $2 \pmod{9}$? Yes!

Isomorphism:

Isomorphisms.

Bijection:

$$f(x) = ax \pmod{m} \text{ if } \gcd(a, m) = 1.$$

Chinese Remainder Theorem: For m, n with $\gcd(n, m) = 1$,
 \implies unique $x \pmod{mn}$ where $x = a \pmod{m}$ and $x = b \pmod{n}$.

Bijection between $(a \pmod{n}, b \pmod{m})$ and $x \pmod{mn}$.

Consider $m = 5, n = 9$, then if $(a, b) = (3, 7)$ then $x = 43 \pmod{45}$.

Consider $(a', b') = (2, 4)$, then $x = 22 \pmod{45}$.

Now consider: $(a, b) + (a', b') = (0, 2)$.

What is x where $x = 0 \pmod{5}$ and $x = 2 \pmod{9}$?

Try $43 + 22 = 65 = 20 \pmod{45}$.

Is it $0 \pmod{5}$? Yes! Is it $2 \pmod{9}$? Yes!

Isomorphism:

the actions $+, \times$ under $(\pmod{5}), (\pmod{9})$

Isomorphisms.

Bijection:

$f(x) = ax \pmod{m}$ if $\gcd(a, m) = 1$.

Chinese Remainder Theorem: For m, n with $\gcd(n, m) = 1$,
 \implies unique $x \pmod{mn}$ where $x = a \pmod{m}$ and $x = b \pmod{n}$.

Bijection between $(a \pmod{n}, b \pmod{m})$ and $x \pmod{mn}$.

Consider $m = 5, n = 9$, then if $(a, b) = (3, 7)$ then $x = 43 \pmod{45}$.

Consider $(a', b') = (2, 4)$, then $x = 22 \pmod{45}$.

Now consider: $(a, b) + (a', b') = (0, 2)$.

What is x where $x = 0 \pmod{5}$ and $x = 2 \pmod{9}$?

Try $43 + 22 = 65 = 20 \pmod{45}$.

Is it $0 \pmod{5}$? Yes! Is it $2 \pmod{9}$? Yes!

Isomorphism:

the actions $+, \times$ under $(\pmod{5}), (\pmod{9})$
correspond to $+, \times$ actions in $(\pmod{45})!$

Public Key Cryptography.

1. Public Key Cryptography
2. RSA system
3. Warnings.

Xor

Computer Science:

Xor

Computer Science:

1 - True

0 - False

Xor

Computer Science:

1 - True

0 - False

$$1 \vee 1 = 1$$

Xor

Computer Science:

1 - True

0 - False

$$1 \vee 1 = 1$$

$$1 \vee 0 = 1$$

$$0 \vee 1 = 1$$

$$0 \vee 0 = 0$$

Xor

Computer Science:

1 - True

0 - False

$$1 \vee 1 = 1$$

$$1 \vee 0 = 1$$

$$0 \vee 1 = 1$$

$$0 \vee 0 = 0$$

$A \oplus B$ - Exclusive or.

Xor

Computer Science:

1 - True

0 - False

$$1 \vee 1 = 1$$

$$1 \vee 0 = 1$$

$$0 \vee 1 = 1$$

$$0 \vee 0 = 0$$

$A \oplus B$ - Exclusive or.

$$1 \oplus 1 = 0$$

Xor

Computer Science:

1 - True

0 - False

$$1 \vee 1 = 1$$

$$1 \vee 0 = 1$$

$$0 \vee 1 = 1$$

$$0 \vee 0 = 0$$

$A \oplus B$ - Exclusive or.

$$1 \oplus 1 = 0$$

$$1 \oplus 0 = 1$$

$$0 \oplus 1 = 1$$

$$0 \oplus 0 = 0$$

Xor

Computer Science:

1 - True

0 - False

$$1 \vee 1 = 1$$

$$1 \vee 0 = 1$$

$$0 \vee 1 = 1$$

$$0 \vee 0 = 0$$

$A \oplus B$ - Exclusive or.

$$1 \oplus 1 = 0$$

$$1 \oplus 0 = 1$$

$$0 \oplus 1 = 1$$

$$0 \oplus 0 = 0$$

Note: Also modular addition modulo 2!

Xor

Computer Science:

1 - True

0 - False

$$1 \vee 1 = 1$$

$$1 \vee 0 = 1$$

$$0 \vee 1 = 1$$

$$0 \vee 0 = 0$$

$A \oplus B$ - Exclusive or.

$$1 \oplus 1 = 0$$

$$1 \oplus 0 = 1$$

$$0 \oplus 1 = 1$$

$$0 \oplus 0 = 0$$

Note: Also modular addition modulo 2!
 $\{0, 1\}$ is set. Take remainder for 2.

Xor

Computer Science:

1 - True

0 - False

$$1 \vee 1 = 1$$

$$1 \vee 0 = 1$$

$$0 \vee 1 = 1$$

$$0 \vee 0 = 0$$

$A \oplus B$ - Exclusive or.

$$1 \oplus 1 = 0$$

$$1 \oplus 0 = 1$$

$$0 \oplus 1 = 1$$

$$0 \oplus 0 = 0$$

Note: Also modular addition modulo 2!
 $\{0, 1\}$ is set. Take remainder for 2.

Xor

Computer Science:

1 - True

0 - False

$$1 \vee 1 = 1$$

$$1 \vee 0 = 1$$

$$0 \vee 1 = 1$$

$$0 \vee 0 = 0$$

$A \oplus B$ - Exclusive or.

$$1 \oplus 1 = 0$$

$$1 \oplus 0 = 1$$

$$0 \oplus 1 = 1$$

$$0 \oplus 0 = 0$$

Note: Also modular addition modulo 2!

$\{0, 1\}$ is set. Take remainder for 2.

Property: $A \oplus B \oplus B = A$.

Xor

Computer Science:

1 - True

0 - False

$$1 \vee 1 = 1$$

$$1 \vee 0 = 1$$

$$0 \vee 1 = 1$$

$$0 \vee 0 = 0$$

$A \oplus B$ - Exclusive or.

$$1 \oplus 1 = 0$$

$$1 \oplus 0 = 1$$

$$0 \oplus 1 = 1$$

$$0 \oplus 0 = 0$$

Note: Also modular addition modulo 2!

$\{0, 1\}$ is set. Take remainder for 2.

Property: $A \oplus B \oplus B = A$.

By cases: $1 \oplus 1 \oplus 1 = 1$.

Xor

Computer Science:

1 - True

0 - False

$$1 \vee 1 = 1$$

$$1 \vee 0 = 1$$

$$0 \vee 1 = 1$$

$$0 \vee 0 = 0$$

$A \oplus B$ - Exclusive or.

$$1 \oplus 1 = 0$$

$$1 \oplus 0 = 1$$

$$0 \oplus 1 = 1$$

$$0 \oplus 0 = 0$$

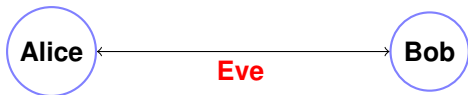
Note: Also modular addition modulo 2!

$\{0, 1\}$ is set. Take remainder for 2.

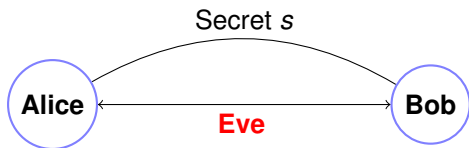
Property: $A \oplus B \oplus B = A$.

By cases: $1 \oplus 1 \oplus 1 = 1$

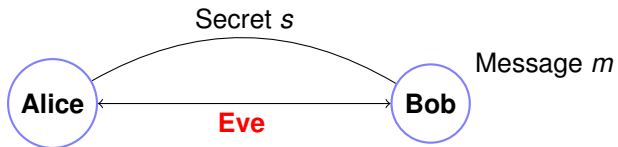
Cryptography ...



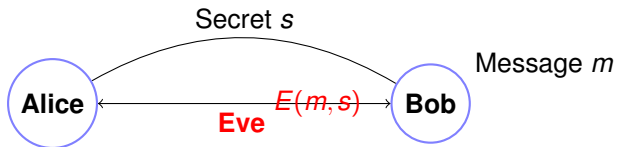
Cryptography ...



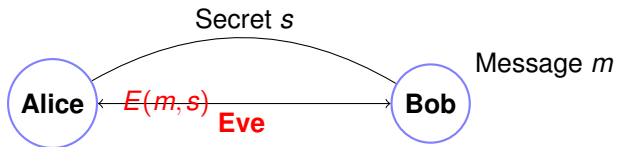
Cryptography ...



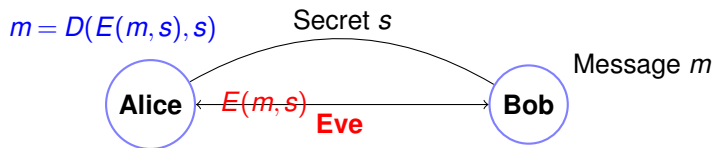
Cryptography ...



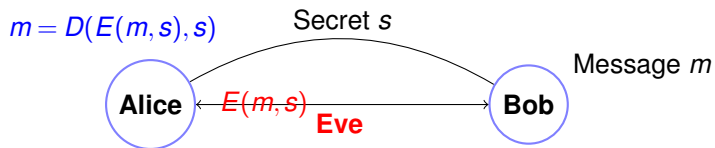
Cryptography ...



Cryptography ...

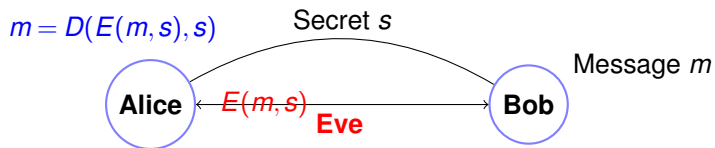


Cryptography ...



Example:

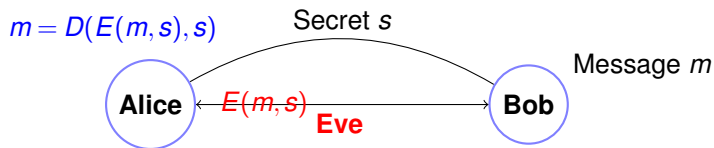
Cryptography ...



Example:

One-time Pad: secret s is string of length $|m|$.

Cryptography ...

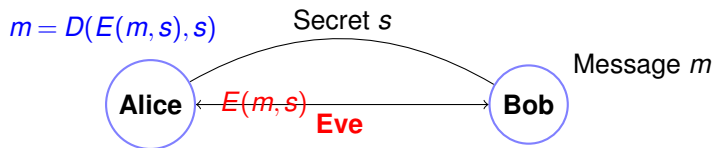


Example:

One-time Pad: secret s is string of length $|m|$.

$m = 10101011110101101$

Cryptography ...



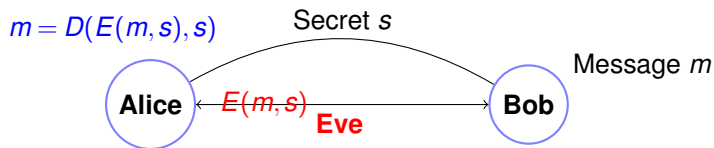
Example:

One-time Pad: secret s is string of length $|m|$.

$m = 10101011110101101$

$s = \dots\dots\dots$

Cryptography ...



Example:

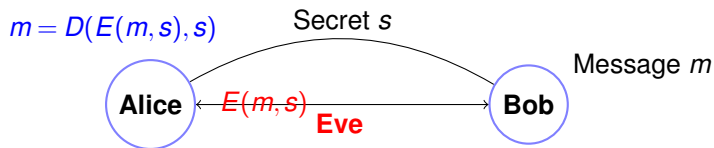
One-time Pad: secret s is string of length $|m|$.

$m = 10101011110101101$

$s = \dots\dots\dots$

$E(m,s)$ – bitwise $m \oplus s$.

Cryptography ...



Example:

One-time Pad: secret s is string of length $|m|$.

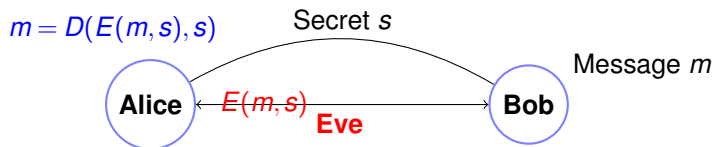
$m = 10101011110101101$

$s = \dots\dots\dots$

$E(m, s)$ – bitwise $m \oplus s$.

$D(x, s)$ – bitwise $x \oplus s$.

Cryptography ...



Example:

One-time Pad: secret s is string of length $|m|$.

$m = 10101011110101101$

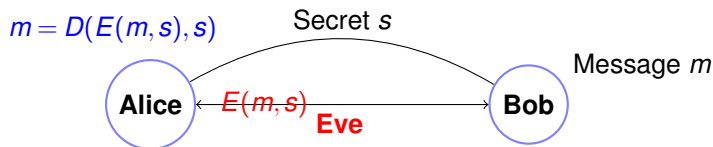
$s = \dots\dots\dots$

$E(m, s)$ – bitwise $m \oplus s$.

$D(x, s)$ – bitwise $x \oplus s$.

Works because $m \oplus s \oplus s = m$!

Cryptography ...



Example:

One-time Pad: secret s is string of length $|m|$.

$m = 10101011110101101$

$s = \dots\dots\dots$

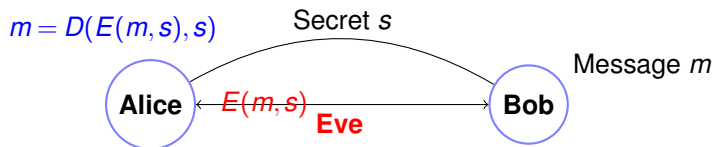
$E(m, s)$ – bitwise $m \oplus s$.

$D(x, s)$ – bitwise $x \oplus s$.

Works because $m \oplus s \oplus s = m$!

...and totally secure!

Cryptography ...



Example:

One-time Pad: secret s is string of length $|m|$.

$m = 10101011110101101$

$s = \dots\dots\dots$

$E(m, s)$ – bitwise $m \oplus s$.

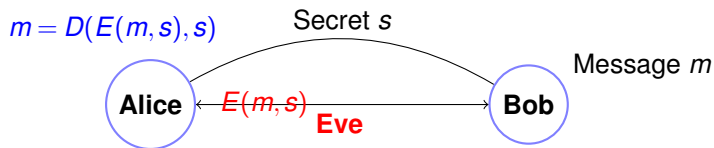
$D(x, s)$ – bitwise $x \oplus s$.

Works because $m \oplus s \oplus s = m$!

...and totally secure!

...given $E(m, s)$ any message m is equally likely.

Cryptography ...



Example:

One-time Pad: secret s is string of length $|m|$.

$m = 10101011110101101$

$s = \dots\dots\dots$

$E(m, s)$ – bitwise $m \oplus s$.

$D(x, s)$ – bitwise $x \oplus s$.

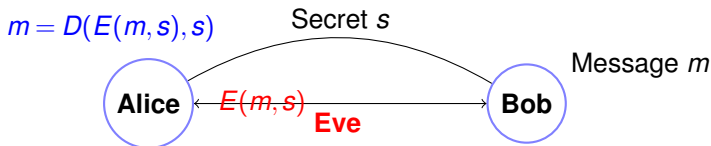
Works because $m \oplus s \oplus s = m$!

...and totally secure!

...given $E(m, s)$ any message m is equally likely.

Disadvantages:

Cryptography ...



Example:

One-time Pad: secret s is string of length $|m|$.

$m = 10101011110101101$

$s = \dots\dots\dots$

$E(m,s)$ – bitwise $m \oplus s$.

$D(x,s)$ – bitwise $x \oplus s$.

Works because $m \oplus s \oplus s = m$!

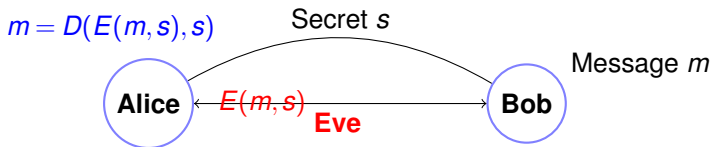
...and totally secure!

...given $E(m,s)$ any message m is equally likely.

Disadvantages:

Shared secret!

Cryptography ...



Example:

One-time Pad: secret s is string of length $|m|$.

$m = 10101011110101101$

$s = \dots\dots\dots$

$E(m, s)$ – bitwise $m \oplus s$.

$D(x, s)$ – bitwise $x \oplus s$.

Works because $m \oplus s \oplus s = m!$

...and totally secure!

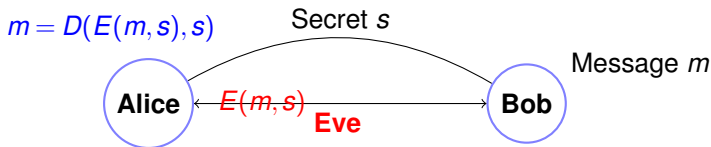
...given $E(m, s)$ any message m is equally likely.

Disadvantages:

Shared secret!

Uses up one time pad..

Cryptography ...



Example:

One-time Pad: secret s is string of length $|m|$.

$m = 10101011110101101$

$s = \dots\dots\dots$

$E(m, s)$ – bitwise $m \oplus s$.

$D(x, s)$ – bitwise $x \oplus s$.

Works because $m \oplus s \oplus s = m!$

...and totally secure!

...given $E(m, s)$ any message m is equally likely.

Disadvantages:

Shared secret!

Uses up one time pad..or less and less secure.

Public key cryptography.



Public key cryptography.



Public key cryptography.

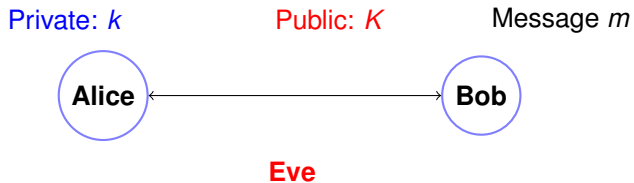
Private: k

Public: K

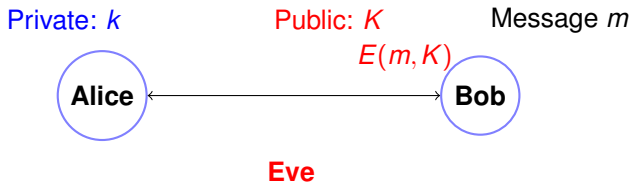


Eve

Public key cryptography.



Public key cryptography.

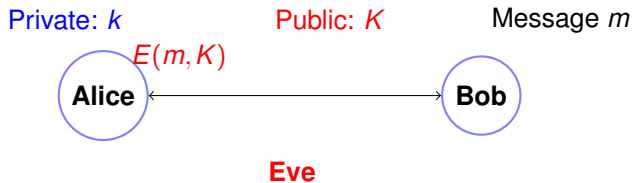


Public key cryptography.



Public key cryptography.

$$m = D(E(m, K), k)$$



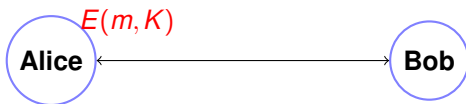
Public key cryptography.

$$m = D(E(m, K), k)$$

Private: k

Public: K

Message m



Everyone knows key K !

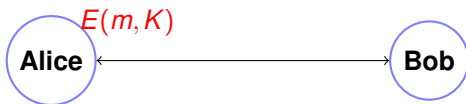
Public key cryptography.

$$m = D(E(m, K), k)$$

Private: k

Public: K

Message m



Everyone knows key K !
Bob (and Eve

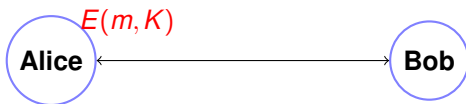
Public key cryptography.

$$m = D(E(m, K), k)$$

Private: k

Public: K

Message m



Everyone knows key K !
Bob (and Eve and me)

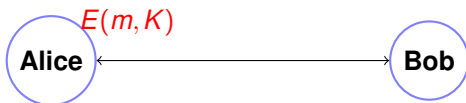
Public key cryptography.

$$m = D(E(m, K), k)$$

Private: k

Public: K

Message m



Eve

Everyone knows key K !
Bob (and Eve and me and you

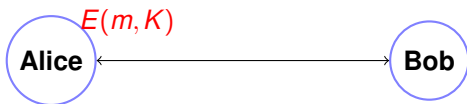
Public key cryptography.

$$m = D(E(m, K), k)$$

Private: k

Public: K

Message m



Eve

Everyone knows key K !

Bob (and Eve and me and you and you ...) can encode.

Public key cryptography.

$$m = D(E(m, K), k)$$



Everyone knows key K !

Bob (and Eve and me and you and ...) can encode.

Only Alice knows the secret key k for public key K .

Public key cryptography.

$$m = D(E(m, K), k)$$



Everyone knows key K !

Bob (and Eve and me and you and ...) can encode.

Only Alice knows the secret key k for public key K .

(Only?) Alice can decode with k .

Public key cryptography.

$$m = D(E(m, K), k)$$



Everyone knows key K !
Bob (and Eve and me and you and ...) can encode.
Only Alice knows the secret key k for public key K .
(Only?) Alice can decode with k .

Is this even possible?

Is public key crypto possible?

²Typically small, say $e = 3$.

Is public key crypto possible?

We don't really know.

²Typically small, say $e = 3$.

Is public key crypto possible?

We don't really know.

...but we do it every day!!!

²Typically small, say $e = 3$.

Is public key crypto possible?

We don't really know.

...but we do it every day!!!

RSA (Rivest, Shamir, and Adleman)

²Typically small, say $e = 3$.

Is public key crypto possible?

We don't really know.

...but we do it every day!!!

RSA (Rivest, Shamir, and Adleman)

Pick two large primes p and q . Let $N = pq$.

²Typically small, say $e = 3$.

Is public key crypto possible?

We don't really know.

...but we do it every day!!!

RSA (Rivest, Shamir, and Adleman)

Pick two large primes p and q . Let $N = pq$.

Choose e relatively prime to $(p-1)(q-1)$.²

²Typically small, say $e = 3$.

Is public key crypto possible?

We don't really know.

...but we do it every day!!!

RSA (Rivest, Shamir, and Adleman)

Pick two large primes p and q . Let $N = pq$.

Choose e relatively prime to $(p-1)(q-1)$.²

Compute $d = e^{-1} \pmod{(p-1)(q-1)}$.

²Typically small, say $e = 3$.

Is public key crypto possible?

We don't really know.

...but we do it every day!!!

RSA (Rivest, Shamir, and Adleman)

Pick two large primes p and q . Let $N = pq$.

Choose e relatively prime to $(p-1)(q-1)$.²

Compute $d = e^{-1} \pmod{(p-1)(q-1)}$.

Announce $N(= p \cdot q)$ and e : $K = (N, e)$ is my public key!

²Typically small, say $e = 3$.

Is public key crypto possible?

We don't really know.

...but we do it every day!!!

RSA (Rivest, Shamir, and Adleman)

Pick two large primes p and q . Let $N = pq$.

Choose e relatively prime to $(p-1)(q-1)$.²

Compute $d = e^{-1} \pmod{(p-1)(q-1)}$.

Announce $N(= p \cdot q)$ and e : $K = (N, e)$ is my public key!

Encoding: $\pmod{(x^e, N)}$.

²Typically small, say $e = 3$.

Is public key crypto possible?

We don't really know.

...but we do it every day!!!

RSA (Rivest, Shamir, and Adleman)

Pick two large primes p and q . Let $N = pq$.

Choose e relatively prime to $(p-1)(q-1)$.²

Compute $d = e^{-1} \pmod{(p-1)(q-1)}$.

Announce $N(= p \cdot q)$ and e : $K = (N, e)$ is my public key!

Encoding: $\text{mod}(x^e, N)$.

Decoding: $\text{mod}(y^d, N)$.

²Typically small, say $e = 3$.

Is public key crypto possible?

We don't really know.

...but we do it every day!!!

RSA (Rivest, Shamir, and Adleman)

Pick two large primes p and q . Let $N = pq$.

Choose e relatively prime to $(p-1)(q-1)$.²

Compute $d = e^{-1} \pmod{(p-1)(q-1)}$.

Announce $N(= p \cdot q)$ and e : $K = (N, e)$ is my public key!

Encoding: $\text{mod}(x^e, N)$.

Decoding: $\text{mod}(y^d, N)$.

Does $D(E(m)) = m^{ed} = m \pmod N$?

²Typically small, say $e = 3$.

Is public key crypto possible?

We don't really know.

...but we do it every day!!!

RSA (Rivest, Shamir, and Adleman)

Pick two large primes p and q . Let $N = pq$.

Choose e relatively prime to $(p-1)(q-1)$.²

Compute $d = e^{-1} \pmod{(p-1)(q-1)}$.

Announce $N(= p \cdot q)$ and e : $K = (N, e)$ is my public key!

Encoding: $\text{mod}(x^e, N)$.

Decoding: $\text{mod}(y^d, N)$.

Does $D(E(m)) = m^{ed} = m \pmod N$?

Yes!

²Typically small, say $e = 3$.

Iterative Extended GCD.

Example: $p = 7$, $q = 11$.

Iterative Extended GCD.

Example: $p = 7$, $q = 11$.

$N = 77$.

Iterative Extended GCD.

Example: $p = 7$, $q = 11$.

$N = 77$.

$$(p - 1)(q - 1) = 60$$

Iterative Extended GCD.

Example: $p = 7$, $q = 11$.

$N = 77$.

$$(p - 1)(q - 1) = 60$$

Choose $e = 7$, since $\gcd(7, 60) = 1$.

Iterative Extended GCD.

Example: $p = 7$, $q = 11$.

$N = 77$.

$$(p - 1)(q - 1) = 60$$

Choose $e = 7$, since $\gcd(7, 60) = 1$.
egcd(7,60).

Iterative Extended GCD.

Example: $p = 7$, $q = 11$.

$N = 77$.

$(p - 1)(q - 1) = 60$

Choose $e = 7$, since $\gcd(7, 60) = 1$.

$e \gcd(7, 60)$.

$$7(0) + 60(1) = 60$$

Iterative Extended GCD.

Example: $p = 7$, $q = 11$.

$N = 77$.

$(p - 1)(q - 1) = 60$

Choose $e = 7$, since $\gcd(7, 60) = 1$.

$e \gcd(7, 60)$.

$$7(0) + 60(1) = 60$$

$$7(1) + 60(0) = 7$$

Iterative Extended GCD.

Example: $p = 7$, $q = 11$.

$N = 77$.

$(p - 1)(q - 1) = 60$

Choose $e = 7$, since $\gcd(7, 60) = 1$.

$e \cdot \text{egcd}(7, 60)$.

$$7(0) + 60(1) = 60$$

$$7(1) + 60(0) = 7$$

$$7(-8) + 60(1) = 4$$

Iterative Extended GCD.

Example: $p = 7$, $q = 11$.

$N = 77$.

$(p - 1)(q - 1) = 60$

Choose $e = 7$, since $\gcd(7, 60) = 1$.
egcd(7,60).

$$7(0) + 60(1) = 60$$

$$7(1) + 60(0) = 7$$

$$7(-8) + 60(1) = 4$$

$$7(9) + 60(-1) = 3$$

Iterative Extended GCD.

Example: $p = 7$, $q = 11$.

$N = 77$.

$(p - 1)(q - 1) = 60$

Choose $e = 7$, since $\gcd(7, 60) = 1$.

$e \gcd(7, 60)$.

$$\begin{aligned}7(0) + 60(1) &= 60 \\7(1) + 60(0) &= 7 \\7(-8) + 60(1) &= 4 \\7(9) + 60(-1) &= 3 \\7(-17) + 60(2) &= 1\end{aligned}$$

Iterative Extended GCD.

Example: $p = 7$, $q = 11$.

$N = 77$.

$(p - 1)(q - 1) = 60$

Choose $e = 7$, since $\gcd(7, 60) = 1$.

$e \gcd(7, 60)$.

$$7(0) + 60(1) = 60$$

$$7(1) + 60(0) = 7$$

$$7(-8) + 60(1) = 4$$

$$7(9) + 60(-1) = 3$$

$$7(-17) + 60(2) = 1$$

Iterative Extended GCD.

Example: $p = 7$, $q = 11$.

$N = 77$.

$(p - 1)(q - 1) = 60$

Choose $e = 7$, since $\gcd(7, 60) = 1$.

$e \gcd(7, 60)$.

$$\begin{aligned}7(0) + 60(1) &= 60 \\7(1) + 60(0) &= 7 \\7(-8) + 60(1) &= 4 \\7(9) + 60(-1) &= 3 \\7(-17) + 60(2) &= 1\end{aligned}$$

Confirm:

Iterative Extended GCD.

Example: $p = 7$, $q = 11$.

$N = 77$.

$(p - 1)(q - 1) = 60$

Choose $e = 7$, since $\gcd(7, 60) = 1$.

$e \gcd(7, 60)$.

$$\begin{aligned}7(0) + 60(1) &= 60 \\7(1) + 60(0) &= 7 \\7(-8) + 60(1) &= 4 \\7(9) + 60(-1) &= 3 \\7(-17) + 60(2) &= 1\end{aligned}$$

Confirm: $-119 + 120 = 1$

Iterative Extended GCD.

Example: $p = 7$, $q = 11$.

$N = 77$.

$(p - 1)(q - 1) = 60$

Choose $e = 7$, since $\gcd(7, 60) = 1$.

$e \gcd(7, 60)$.

$$\begin{aligned}7(0) + 60(1) &= 60 \\7(1) + 60(0) &= 7 \\7(-8) + 60(1) &= 4 \\7(9) + 60(-1) &= 3 \\7(-17) + 60(2) &= 1\end{aligned}$$

Confirm: $-119 + 120 = 1$

$d = e^{-1} = -17 = 43 = \pmod{60}$

Encryption/Decryption Techniques.

Encryption/Decryption Techniques.

Public Key: (77, 7)

Encryption/Decryption Techniques.

Public Key: $(77, 7)$

Message Choices: $\{0, \dots, 76\}$.

Encryption/Decryption Techniques.

Public Key: $(77, 7)$

Message Choices: $\{0, \dots, 76\}$.

Message: 2!

Encryption/Decryption Techniques.

Public Key: $(77, 7)$

Message Choices: $\{0, \dots, 76\}$.

Message: 2!

$E(2)$

Encryption/Decryption Techniques.

Public Key: (77, 7)

Message Choices: $\{0, \dots, 76\}$.

Message: 2!

$$E(2) = 2^e$$

Encryption/Decryption Techniques.

Public Key: (77, 7)

Message Choices: $\{0, \dots, 76\}$.

Message: 2!

$$E(2) = 2^e = 2^7$$

Encryption/Decryption Techniques.

Public Key: $(77, 7)$

Message Choices: $\{0, \dots, 76\}$.

Message: 2!

$$E(2) = 2^e = 2^7 \equiv 128 \pmod{77}$$

Encryption/Decryption Techniques.

Public Key: $(77, 7)$

Message Choices: $\{0, \dots, 76\}$.

Message: 2!

$$E(2) = 2^e = 2^7 \equiv 128 \pmod{77} = 51 \pmod{77}$$

Encryption/Decryption Techniques.

Public Key: $(77, 7)$

Message Choices: $\{0, \dots, 76\}$.

Message: 2!

$$E(2) = 2^e = 2^7 \equiv 128 \pmod{77} = 51 \pmod{77}$$

$$D(51) = 51^{43} \pmod{77}$$

Encryption/Decryption Techniques.

Public Key: $(77, 7)$

Message Choices: $\{0, \dots, 76\}$.

Message: 2!

$$E(2) = 2^e = 2^7 \equiv 128 \pmod{77} = 51 \pmod{77}$$

$$D(51) = 51^{43} \pmod{77}$$

uh oh!

Encryption/Decryption Techniques.

Public Key: $(77, 7)$

Message Choices: $\{0, \dots, 76\}$.

Message: 2!

$$E(2) = 2^e = 2^7 \equiv 128 \pmod{77} = 51 \pmod{77}$$

$$D(51) = 51^{43} \pmod{77}$$

uh oh!

Obvious way: 43 multiplications.

Encryption/Decryption Techniques.

Public Key: $(77, 7)$

Message Choices: $\{0, \dots, 76\}$.

Message: 2!

$$E(2) = 2^e = 2^7 \equiv 128 \pmod{77} = 51 \pmod{77}$$

$$D(51) = 51^{43} \pmod{77}$$

uh oh!

Obvious way: 43 multiplications. **Ouch.**

Encryption/Decryption Techniques.

Public Key: $(77, 7)$

Message Choices: $\{0, \dots, 76\}$.

Message: 2!

$$E(2) = 2^e = 2^7 \equiv 128 \pmod{77} = 51 \pmod{77}$$

$$D(51) = 51^{43} \pmod{77}$$

uh oh!

Obvious way: 43 multiplications. **Ouch.**

In general, $O(N)$ or $O(2^n)$ multiplications!

Repeated squaring.

Repeated squaring.

Notice: $43 = 32 + 8 + 2 + 1$.

Repeated squaring.

Notice: $43 = 32 + 8 + 2 + 1$. 51^{43}

Repeated squaring.

Notice: $43 = 32 + 8 + 2 + 1$. $51^{43} = 51^{32+8+2+1}$

Repeated squaring.

Notice: $43 = 32 + 8 + 2 + 1$. $51^{43} = 51^{32+8+2+1} = 51^{32} \cdot 51^8 \cdot 51^2 \cdot 51^1$
(mod 77).

Repeated squaring.

Notice: $43 = 32 + 8 + 2 + 1$. $51^{43} = 51^{32+8+2+1} = 51^{32} \cdot 51^8 \cdot 51^2 \cdot 51^1$
(mod 77).

4 multiplications sort of...

Repeated squaring.

Notice: $43 = 32 + 8 + 2 + 1$. $51^{43} = 51^{32+8+2+1} = 51^{32} \cdot 51^8 \cdot 51^2 \cdot 51^1$
(mod 77).

4 multiplications sort of...

Need to compute $51^{32} \dots 51^1$.?

Repeated squaring.

Notice: $43 = 32 + 8 + 2 + 1$. $51^{43} \equiv 51^{32+8+2+1} \equiv 51^{32} \cdot 51^8 \cdot 51^2 \cdot 51^1$
(mod 77).

4 multiplications sort of...

Need to compute $51^{32} \dots 51^1$?

$51^1 \equiv 51 \pmod{77}$

Repeated squaring.

Notice: $43 = 32 + 8 + 2 + 1$. $51^{43} \equiv 51^{32+8+2+1} \equiv 51^{32} \cdot 51^8 \cdot 51^2 \cdot 51^1$
(mod 77).

4 multiplications sort of...

Need to compute $51^{32} \dots 51^1$?

$$51^1 \equiv 51 \pmod{77}$$

$$51^2 =$$

Repeated squaring.

Notice: $43 = 32 + 8 + 2 + 1$. $51^{43} = 51^{32+8+2+1} = 51^{32} \cdot 51^8 \cdot 51^2 \cdot 51^1$
(mod 77).

4 multiplications sort of...

Need to compute $51^{32} \dots 51^1$.?

$$51^1 \equiv 51 \pmod{77}$$

$$51^2 = (51) * (51) = 2601 \equiv 60 \pmod{77}$$

Repeated squaring.

Notice: $43 = 32 + 8 + 2 + 1$. $51^{43} = 51^{32+8+2+1} = 51^{32} \cdot 51^8 \cdot 51^2 \cdot 51^1$
(mod 77).

4 multiplications sort of...

Need to compute $51^{32} \dots 51^1$.?

$$51^1 \equiv 51 \pmod{77}$$

$$51^2 = (51) * (51) = 2601 \equiv 60 \pmod{77}$$

$$51^4 =$$

Repeated squaring.

Notice: $43 = 32 + 8 + 2 + 1$. $51^{43} = 51^{32+8+2+1} = 51^{32} \cdot 51^8 \cdot 51^2 \cdot 51^1$
(mod 77).

4 multiplications sort of...

Need to compute $51^{32} \dots 51^1$.?

$$51^1 \equiv 51 \pmod{77}$$

$$51^2 = (51) * (51) = 2601 \equiv 60 \pmod{77}$$

$$51^4 = (51^2) * (51^2)$$

Repeated squaring.

Notice: $43 = 32 + 8 + 2 + 1$. $51^{43} = 51^{32+8+2+1} = 51^{32} \cdot 51^8 \cdot 51^2 \cdot 51^1$
(mod 77).

4 multiplications sort of...

Need to compute $51^{32} \dots 51^1$.?

$$51^1 \equiv 51 \pmod{77}$$

$$51^2 = (51) * (51) = 2601 \equiv 60 \pmod{77}$$

$$51^4 = (51^2) * (51^2) = 60 * 60 = 3600 \equiv 58 \pmod{77}$$

Repeated squaring.

Notice: $43 = 32 + 8 + 2 + 1$. $51^{43} = 51^{32+8+2+1} = 51^{32} \cdot 51^8 \cdot 51^2 \cdot 51^1$
(mod 77).

4 multiplications sort of...

Need to compute $51^{32} \dots 51^1$.?

$$51^1 \equiv 51 \pmod{77}$$

$$51^2 = (51) * (51) = 2601 \equiv 60 \pmod{77}$$

$$51^4 = (51^2) * (51^2) = 60 * 60 = 3600 \equiv 58 \pmod{77}$$

$$51^8 =$$

Repeated squaring.

Notice: $43 = 32 + 8 + 2 + 1$. $51^{43} = 51^{32+8+2+1} = 51^{32} \cdot 51^8 \cdot 51^2 \cdot 51^1$
(mod 77).

4 multiplications sort of...

Need to compute $51^{32} \dots 51^1$.?

$$51^1 \equiv 51 \pmod{77}$$

$$51^2 = (51) * (51) = 2601 \equiv 60 \pmod{77}$$

$$51^4 = (51^2) * (51^2) = 60 * 60 = 3600 \equiv 58 \pmod{77}$$

$$51^8 = (51^4) * (51^4)$$

Repeated squaring.

Notice: $43 = 32 + 8 + 2 + 1$. $51^{43} = 51^{32+8+2+1} = 51^{32} \cdot 51^8 \cdot 51^2 \cdot 51^1$
(mod 77).

4 multiplications sort of...

Need to compute $51^{32} \dots 51^1$.?

$$51^1 \equiv 51 \pmod{77}$$

$$51^2 = (51) * (51) = 2601 \equiv 60 \pmod{77}$$

$$51^4 = (51^2) * (51^2) = 60 * 60 = 3600 \equiv 58 \pmod{77}$$

$$51^8 = (51^4) * (51^4) = 58 * 58 = 3364 \equiv 53 \pmod{77}$$

Repeated squaring.

Notice: $43 = 32 + 8 + 2 + 1$. $51^{43} = 51^{32+8+2+1} = 51^{32} \cdot 51^8 \cdot 51^2 \cdot 51^1$
(mod 77).

4 multiplications sort of...

Need to compute $51^{32} \dots 51^1$.?

$$51^1 \equiv 51 \pmod{77}$$

$$51^2 = (51) * (51) = 2601 \equiv 60 \pmod{77}$$

$$51^4 = (51^2) * (51^2) = 60 * 60 = 3600 \equiv 58 \pmod{77}$$

$$51^8 = (51^4) * (51^4) = 58 * 58 = 3364 \equiv 53 \pmod{77}$$

$$51^{16} = (51^8) * (51^8) = 53 * 53 = 2809 \equiv 37 \pmod{77}$$

Repeated squaring.

Notice: $43 = 32 + 8 + 2 + 1$. $51^{43} = 51^{32+8+2+1} = 51^{32} \cdot 51^8 \cdot 51^2 \cdot 51^1$
(mod 77).

4 multiplications sort of...

Need to compute $51^{32} \dots 51^1$.?

$$51^1 \equiv 51 \pmod{77}$$

$$51^2 = (51) * (51) = 2601 \equiv 60 \pmod{77}$$

$$51^4 = (51^2) * (51^2) = 60 * 60 = 3600 \equiv 58 \pmod{77}$$

$$51^8 = (51^4) * (51^4) = 58 * 58 = 3364 \equiv 53 \pmod{77}$$

$$51^{16} = (51^8) * (51^8) = 53 * 53 = 2809 \equiv 37 \pmod{77}$$

$$51^{32} = (51^{16}) * (51^{16}) = 37 * 37 = 1369 \equiv 60 \pmod{77}$$

Repeated squaring.

Notice: $43 = 32 + 8 + 2 + 1$. $51^{43} = 51^{32+8+2+1} = 51^{32} \cdot 51^8 \cdot 51^2 \cdot 51^1$
(mod 77).

4 multiplications sort of...

Need to compute $51^{32} \dots 51^1$.?

$$51^1 \equiv 51 \pmod{77}$$

$$51^2 = (51) * (51) = 2601 \equiv 60 \pmod{77}$$

$$51^4 = (51^2) * (51^2) = 60 * 60 = 3600 \equiv 58 \pmod{77}$$

$$51^8 = (51^4) * (51^4) = 58 * 58 = 3364 \equiv 53 \pmod{77}$$

$$51^{16} = (51^8) * (51^8) = 53 * 53 = 2809 \equiv 37 \pmod{77}$$

$$51^{32} = (51^{16}) * (51^{16}) = 37 * 37 = 1369 \equiv 60 \pmod{77}$$

5 more multiplications.

Repeated squaring.

Notice: $43 = 32 + 8 + 2 + 1$. $51^{43} = 51^{32+8+2+1} = 51^{32} \cdot 51^8 \cdot 51^2 \cdot 51^1$
(mod 77).

4 multiplications sort of...

Need to compute $51^{32} \dots 51^1$.?

$$51^1 \equiv 51 \pmod{77}$$

$$51^2 = (51) * (51) = 2601 \equiv 60 \pmod{77}$$

$$51^4 = (51^2) * (51^2) = 60 * 60 = 3600 \equiv 58 \pmod{77}$$

$$51^8 = (51^4) * (51^4) = 58 * 58 = 3364 \equiv 53 \pmod{77}$$

$$51^{16} = (51^8) * (51^8) = 53 * 53 = 2809 \equiv 37 \pmod{77}$$

$$51^{32} = (51^{16}) * (51^{16}) = 37 * 37 = 1369 \equiv 60 \pmod{77}$$

5 more multiplications.

$$51^{32} \cdot 51^8 \cdot 51^2 \cdot 51^1 = (60) * (53) * (60) * (51) \equiv 2 \pmod{77}.$$

Repeated squaring.

Notice: $43 = 32 + 8 + 2 + 1$. $51^{43} = 51^{32+8+2+1} = 51^{32} \cdot 51^8 \cdot 51^2 \cdot 51^1$
(mod 77).

4 multiplications sort of...

Need to compute $51^{32} \dots 51^1$.?

$$51^1 \equiv 51 \pmod{77}$$

$$51^2 = (51) * (51) = 2601 \equiv 60 \pmod{77}$$

$$51^4 = (51^2) * (51^2) = 60 * 60 = 3600 \equiv 58 \pmod{77}$$

$$51^8 = (51^4) * (51^4) = 58 * 58 = 3364 \equiv 53 \pmod{77}$$

$$51^{16} = (51^8) * (51^8) = 53 * 53 = 2809 \equiv 37 \pmod{77}$$

$$51^{32} = (51^{16}) * (51^{16}) = 37 * 37 = 1369 \equiv 60 \pmod{77}$$

5 more multiplications.

$$51^{32} \cdot 51^8 \cdot 51^2 \cdot 51^1 = (60) * (53) * (60) * (51) \equiv 2 \pmod{77}.$$

Decoding got the message back!

Repeated squaring.

Notice: $43 = 32 + 8 + 2 + 1$. $51^{43} = 51^{32+8+2+1} = 51^{32} \cdot 51^8 \cdot 51^2 \cdot 51^1$
(mod 77).

4 multiplications sort of...

Need to compute $51^{32} \dots 51^1$.?

$$51^1 \equiv 51 \pmod{77}$$

$$51^2 = (51) * (51) = 2601 \equiv 60 \pmod{77}$$

$$51^4 = (51^2) * (51^2) = 60 * 60 = 3600 \equiv 58 \pmod{77}$$

$$51^8 = (51^4) * (51^4) = 58 * 58 = 3364 \equiv 53 \pmod{77}$$

$$51^{16} = (51^8) * (51^8) = 53 * 53 = 2809 \equiv 37 \pmod{77}$$

$$51^{32} = (51^{16}) * (51^{16}) = 37 * 37 = 1369 \equiv 60 \pmod{77}$$

5 more multiplications.

$$51^{32} \cdot 51^8 \cdot 51^2 \cdot 51^1 = (60) * (53) * (60) * (51) \equiv 2 \pmod{77}.$$

Decoding got the message back!

Repeated Squaring took 9 multiplications

Repeated squaring.

Notice: $43 = 32 + 8 + 2 + 1$. $51^{43} = 51^{32+8+2+1} = 51^{32} \cdot 51^8 \cdot 51^2 \cdot 51^1$
(mod 77).

4 multiplications sort of...

Need to compute $51^{32} \dots 51^1$.?

$$51^1 \equiv 51 \pmod{77}$$

$$51^2 = (51) * (51) = 2601 \equiv 60 \pmod{77}$$

$$51^4 = (51^2) * (51^2) = 60 * 60 = 3600 \equiv 58 \pmod{77}$$

$$51^8 = (51^4) * (51^4) = 58 * 58 = 3364 \equiv 53 \pmod{77}$$

$$51^{16} = (51^8) * (51^8) = 53 * 53 = 2809 \equiv 37 \pmod{77}$$

$$51^{32} = (51^{16}) * (51^{16}) = 37 * 37 = 1369 \equiv 60 \pmod{77}$$

5 more multiplications.

$$51^{32} \cdot 51^8 \cdot 51^2 \cdot 51^1 = (60) * (53) * (60) * (51) \equiv 2 \pmod{77}.$$

Decoding got the message back!

Repeated Squaring took 9 multiplications versus 43.

Recursive version.

```
(define (power x y m)
  (if (= y 1)
      (mod x m)
      (let ((x-to-evened-y (power (square x) (/ y 2) m)))
        (if (evenp y)
            x-to-evened-y
            (mod (* x x-to-evened-y) m ))))))
```

Claim: Program correctly computes x^y .

Recursive version.

```
(define (power x y m)
  (if (= y 1)
      (mod x m)
      (let ((x-to-evened-y (power (square x) (/ y 2) m)))
        (if (evenp y)
            x-to-evened-y
            (mod (* x x-to-evened-y) m ))))))
```

Claim: Program correctly computes x^y .

Base: $x^1 = x \pmod{m}$.

Recursive version.

```
(define (power x y m)
  (if (= y 1)
      (mod x m)
      (let ((x-to-evened-y (power (square x) (/ y 2) m)))
        (if (evenp y)
            x-to-evened-y
            (mod (* x x-to-evened-y) m ))))))
```

Claim: Program correctly computes x^y .

Base: $x^1 = x \pmod{m}$.

$$x^y = x^{2(y/2) + \text{mod}(y,2)} = (x^2)^{y/2} x^{\text{mod} 2} \pmod{m}.$$

Recursive version.

```
(define (power x y m)
  (if (= y 1)
      (mod x m)
      (let ((x-to-evened-y (power (square x) (/ y 2) m)))
        (if (evenp y)
            x-to-evened-y
            (mod (* x x-to-evened-y) m ))))))
```

Claim: Program correctly computes x^y .

Base: $x^1 = x \pmod m$.

$$x^y = x^{2(y/2) + \text{mod}(y,2)} = (x^2)^{y/2} x^{\text{mod} 2} \pmod m.$$

Last expression computed in recursive call with x^2 and $y/2$.

Recursive version.

```
(define (power x y m)
  (if (= y 1)
      (mod x m)
      (let ((x-to-evened-y (power (square x) (/ y 2) m)))
        (if (evenp y)
            x-to-evened-y
            (mod (* x x-to-evened-y) m ))))))
```

Claim: Program correctly computes x^y .

Base: $x^1 = x \pmod{m}$.

$$x^y = x^{2(y/2) + \text{mod}(y,2)} = (x^2)^{y/2} x^{\text{mod} 2} \pmod{m}.$$

Last expression computed in recursive call with x^2 and $y/2$.

Note: $y/2$ is integer division.

Repeated Squaring: x^y

Repeated Squaring: x^y

Repeated squaring $O(\log y)$ multiplications versus $y!!!$

1. x^y : Compute x^1 ,

Repeated Squaring: x^y

Repeated squaring $O(\log y)$ multiplications versus $y!!!$

1. x^y : Compute $x^1, x^2,$

Repeated Squaring: x^y

Repeated squaring $O(\log y)$ multiplications versus $y!!!$

1. x^y : Compute $x^1, x^2, x^4,$

Repeated Squaring: x^y

Repeated squaring $O(\log y)$ multiplications versus $y!!!$

1. x^y : Compute $x^1, x^2, x^4, \dots,$

Repeated Squaring: x^y

Repeated squaring $O(\log y)$ multiplications versus $y!!!$

1. x^y : Compute $x^1, x^2, x^4, \dots, x^{2^{\lceil \log y \rceil}}$.

Repeated Squaring: x^y

Repeated squaring $O(\log y)$ multiplications versus $y!!!$

1. x^y : Compute $x^1, x^2, x^4, \dots, x^{2^{\lfloor \log y \rfloor}}$.
2. Multiply together x^i where the (i) th bit of y (in binary) is 1.

Repeated Squaring: x^y

Repeated squaring $O(\log y)$ multiplications versus $y!!!$

1. x^y : Compute $x^1, x^2, x^4, \dots, x^{2^{\lfloor \log y \rfloor}}$.
2. Multiply together x^i where the $(\log(i))$ th bit of y (in binary) is 1.
Example:

Repeated Squaring: x^y

Repeated squaring $O(\log y)$ multiplications versus $y!!!$

1. x^y : Compute $x^1, x^2, x^4, \dots, x^{2^{\lfloor \log y \rfloor}}$.
2. Multiply together x^i where the (i) th bit of y (in binary) is 1.
Example: $43 = 101011$ in binary.

Repeated Squaring: x^y

Repeated squaring $O(\log y)$ multiplications versus $y!!!$

1. x^y : Compute $x^1, x^2, x^4, \dots, x^{2^{\lfloor \log y \rfloor}}$.
2. Multiply together x^i where the $(\log(i))$ th bit of y (in binary) is 1.

Example: $43 = 101011$ in binary.

$$x^{43} = x^{32} * x^8 * x^2 * x^1.$$

Repeated Squaring: x^y

Repeated squaring $O(\log y)$ multiplications versus $y!!!$

1. x^y : Compute $x^1, x^2, x^4, \dots, x^{2^{\lceil \log y \rceil}}$.
2. Multiply together x^i where the $(\log(i))$ th bit of y (in binary) is 1.
Example: $43 = 101011$ in binary.
$$x^{43} = x^{32} * x^8 * x^2 * x^1.$$

Modular Exponentiation: $x^y \pmod N$.

Repeated Squaring: x^y

Repeated squaring $O(\log y)$ multiplications versus $y!!!$

1. x^y : Compute $x^1, x^2, x^4, \dots, x^{2^{\lceil \log y \rceil}}$.
2. Multiply together x^i where the $(\log(i))$ th bit of y (in binary) is 1.
Example: $43 = 101011$ in binary.
$$x^{43} = x^{32} * x^8 * x^2 * x^1.$$

Modular Exponentiation: $x^y \pmod N$. All n -bit numbers.

Repeated Squaring: x^y

Repeated squaring $O(\log y)$ multiplications versus $y!!!$

1. x^y : Compute $x^1, x^2, x^4, \dots, x^{2^{\lceil \log y \rceil}}$.
2. Multiply together x^i where the $(\log(i))$ th bit of y (in binary) is 1.
Example: $43 = 101011$ in binary.
$$x^{43} = x^{32} * x^8 * x^2 * x^1.$$

Modular Exponentiation: $x^y \pmod N$. All n -bit numbers.

Repeated Squaring:

Repeated Squaring: x^y

Repeated squaring $O(\log y)$ multiplications versus $y!!!$

1. x^y : Compute $x^1, x^2, x^4, \dots, x^{2^{\lfloor \log y \rfloor}}$.
2. Multiply together x^i where the $(\log(i))$ th bit of y (in binary) is 1.
Example: $43 = 101011$ in binary.
$$x^{43} = x^{32} * x^8 * x^2 * x^1.$$

Modular Exponentiation: $x^y \pmod N$. All n -bit numbers.

Repeated Squaring:
 $O(n)$ multiplications.

Repeated Squaring: x^y

Repeated squaring $O(\log y)$ multiplications versus $y!!!$

1. x^y : Compute $x^1, x^2, x^4, \dots, x^{2^{\lceil \log y \rceil}}$.
2. Multiply together x^i where the $(\log(i))$ th bit of y (in binary) is 1.
Example: $43 = 101011$ in binary.
$$x^{43} = x^{32} * x^8 * x^2 * x^1.$$

Modular Exponentiation: $x^y \pmod N$. All n -bit numbers.

Repeated Squaring:

$O(n)$ multiplications.

$O(n^2)$ time per multiplication.

Repeated Squaring: x^y

Repeated squaring $O(\log y)$ multiplications versus $y!!!$

1. x^y : Compute $x^1, x^2, x^4, \dots, x^{2^{\lfloor \log y \rfloor}}$.
2. Multiply together x^i where the $(\log(i))$ th bit of y (in binary) is 1.
Example: $43 = 101011$ in binary.
$$x^{43} = x^{32} * x^8 * x^2 * x^1.$$

Modular Exponentiation: $x^y \pmod N$. All n -bit numbers.

Repeated Squaring:

$O(n)$ multiplications.

$O(n^2)$ time per multiplication.

$\implies O(n^3)$ time.

Conclusion: $x^y \pmod N$

Repeated Squaring: x^y

Repeated squaring $O(\log y)$ multiplications versus $y!!!$

1. x^y : Compute $x^1, x^2, x^4, \dots, x^{2^{\lfloor \log y \rfloor}}$.
2. Multiply together x^i where the $(\log(i))$ th bit of y (in binary) is 1.
Example: $43 = 101011$ in binary.
$$x^{43} = x^{32} * x^8 * x^2 * x^1.$$

Modular Exponentiation: $x^y \pmod N$. All n -bit numbers.

Repeated Squaring:

$O(n)$ multiplications.

$O(n^2)$ time per multiplication.

$\implies O(n^3)$ time.

Conclusion: $x^y \pmod N$ takes $O(n^3)$ time.

RSA is pretty fast.

Modular Exponentiation: $x^y \pmod N$.

RSA is pretty fast.

Modular Exponentiation: $x^y \pmod N$. All n -bit numbers.
 $O(n^3)$ time.

RSA is pretty fast.

Modular Exponentiation: $x^y \pmod N$. All n -bit numbers.
 $O(n^3)$ time.

Remember RSA encoding/decoding!

RSA is pretty fast.

Modular Exponentiation: $x^y \pmod N$. All n -bit numbers.
 $O(n^3)$ time.

Remember RSA encoding/decoding!

$$E(m, (N, e)) = m^e \pmod N.$$

RSA is pretty fast.

Modular Exponentiation: $x^y \pmod N$. All n -bit numbers.
 $O(n^3)$ time.

Remember RSA encoding/decoding!

$$E(m, (N, e)) = m^e \pmod N.$$

$$D(m, (N, d)) = m^d \pmod N.$$

RSA is pretty fast.

Modular Exponentiation: $x^y \pmod N$. All n -bit numbers.
 $O(n^3)$ time.

Remember RSA encoding/decoding!

$$E(m, (N, e)) = m^e \pmod N.$$

$$D(m, (N, d)) = m^d \pmod N.$$

RSA is pretty fast.

Modular Exponentiation: $x^y \pmod N$. All n -bit numbers.
 $O(n^3)$ time.

Remember RSA encoding/decoding!

$$E(m, (N, e)) = m^e \pmod N.$$

$$D(m, (N, d)) = m^d \pmod N.$$

For 512 bits, a few hundred million operations.

RSA is pretty fast.

Modular Exponentiation: $x^y \pmod N$. All n -bit numbers.
 $O(n^3)$ time.

Remember RSA encoding/decoding!

$$E(m, (N, e)) = m^e \pmod N.$$

$$D(m, (N, d)) = m^d \pmod N.$$

For 512 bits, a few hundred million operations.

Easy, peasey.

Decoding.

$$E(m, (N, e)) = m^e \pmod{N}.$$

Decoding.

$$E(m, (N, e)) = m^e \pmod{N}.$$

$$D(m, (N, d)) = m^d \pmod{N}.$$

Decoding.

$$E(m, (N, e)) = m^e \pmod{N}.$$

$$D(m, (N, d)) = m^d \pmod{N}.$$

Decoding.

$$E(m, (N, e)) = m^e \pmod{N}.$$

$$D(m, (N, d)) = m^d \pmod{N}.$$

$$N = pq$$

Decoding.

$$E(m, (N, e)) = m^e \pmod{N}.$$

$$D(m, (N, d)) = m^d \pmod{N}.$$

$$N = pq \text{ and } d = e^{-1} \pmod{(p-1)(q-1)}.$$

Decoding.

$$E(m, (N, e)) = m^e \pmod{N}.$$

$$D(m, (N, d)) = m^d \pmod{N}.$$

$$N = pq \text{ and } d = e^{-1} \pmod{(p-1)(q-1)}.$$

Want:

Decoding.

$$E(m, (N, e)) = m^e \pmod{N}.$$

$$D(m, (N, d)) = m^d \pmod{N}.$$

$$N = pq \text{ and } d = e^{-1} \pmod{(p-1)(q-1)}.$$

$$\text{Want: } (m^e)^d = m^{ed} = m \pmod{N}.$$

Always decode correctly?

$$E(m, (N, e)) = m^e \pmod{N}.$$

Always decode correctly?

$$E(m, (N, e)) = m^e \pmod{N}.$$

$$D(m, (N, d)) = m^d \pmod{N}.$$

Always decode correctly?

$$E(m, (N, e)) = m^e \pmod{N}.$$

$$D(m, (N, d)) = m^d \pmod{N}.$$

Always decode correctly?

$$E(m, (N, e)) = m^e \pmod{N}.$$

$$D(m, (N, d)) = m^d \pmod{N}.$$

$$N = pq$$

Always decode correctly?

$$E(m, (N, e)) = m^e \pmod{N}.$$

$$D(m, (N, d)) = m^d \pmod{N}.$$

$$N = pq \text{ and } d = e^{-1} \pmod{(p-1)(q-1)}.$$

Always decode correctly?

$$E(m, (N, e)) = m^e \pmod{N}.$$

$$D(m, (N, d)) = m^d \pmod{N}.$$

$$N = pq \text{ and } d = e^{-1} \pmod{(p-1)(q-1)}.$$

Want:

Always decode correctly?

$$E(m, (N, e)) = m^e \pmod{N}.$$

$$D(m, (N, d)) = m^d \pmod{N}.$$

$$N = pq \text{ and } d = e^{-1} \pmod{(p-1)(q-1)}.$$

$$\text{Want: } (m^e)^d = m^{ed} = m \pmod{N}.$$

Always decode correctly?

$$E(m, (N, e)) = m^e \pmod{N}.$$

$$D(m, (N, d)) = m^d \pmod{N}.$$

$$N = pq \text{ and } d = e^{-1} \pmod{(p-1)(q-1)}.$$

$$\text{Want: } (m^e)^d = m^{ed} = m \pmod{N}.$$

Another view:

Always decode correctly?

$$E(m, (N, e)) = m^e \pmod{N}.$$

$$D(m, (N, d)) = m^d \pmod{N}.$$

$$N = pq \text{ and } d = e^{-1} \pmod{(p-1)(q-1)}.$$

$$\text{Want: } (m^e)^d = m^{ed} = m \pmod{N}.$$

Another view:

$$d = e^{-1} \pmod{(p-1)(q-1)} \iff ed = k(p-1)(q-1) + 1.$$

Always decode correctly?

$$E(m, (N, e)) = m^e \pmod{N}.$$

$$D(m, (N, d)) = m^d \pmod{N}.$$

$$N = pq \text{ and } d = e^{-1} \pmod{(p-1)(q-1)}.$$

$$\text{Want: } (m^e)^d = m^{ed} = m \pmod{N}.$$

Another view:

$$d = e^{-1} \pmod{(p-1)(q-1)} \iff ed = k(p-1)(q-1) + 1.$$

Consider...

Always decode correctly?

$$E(m, (N, e)) = m^e \pmod{N}.$$

$$D(m, (N, d)) = m^d \pmod{N}.$$

$$N = pq \text{ and } d = e^{-1} \pmod{(p-1)(q-1)}.$$

$$\text{Want: } (m^e)^d = m^{ed} = m \pmod{N}.$$

Another view:

$$d = e^{-1} \pmod{(p-1)(q-1)} \iff ed = k(p-1)(q-1) + 1.$$

Consider...

Fermat's Little Theorem: For prime p , and $a \not\equiv 0 \pmod{p}$,

Always decode correctly?

$$E(m, (N, e)) = m^e \pmod{N}.$$

$$D(m, (N, d)) = m^d \pmod{N}.$$

$$N = pq \text{ and } d = e^{-1} \pmod{(p-1)(q-1)}.$$

$$\text{Want: } (m^e)^d = m^{ed} = m \pmod{N}.$$

Another view:

$$d = e^{-1} \pmod{(p-1)(q-1)} \iff ed = k(p-1)(q-1) + 1.$$

Consider...

Fermat's Little Theorem: For prime p , and $a \not\equiv 0 \pmod{p}$,

$$a^{p-1} \equiv 1 \pmod{p}.$$

Always decode correctly?

$$E(m, (N, e)) = m^e \pmod{N}.$$

$$D(m, (N, d)) = m^d \pmod{N}.$$

$$N = pq \text{ and } d = e^{-1} \pmod{(p-1)(q-1)}.$$

$$\text{Want: } (m^e)^d = m^{ed} = m \pmod{N}.$$

Another view:

$$d = e^{-1} \pmod{(p-1)(q-1)} \iff ed = k(p-1)(q-1) + 1.$$

Consider...

Fermat's Little Theorem: For prime p , and $a \not\equiv 0 \pmod{p}$,

$$a^{p-1} \equiv 1 \pmod{p}.$$

$$\implies a^{k(p-1)} \equiv 1 \pmod{p}$$

Always decode correctly?

$$E(m, (N, e)) = m^e \pmod{N}.$$

$$D(m, (N, d)) = m^d \pmod{N}.$$

$$N = pq \text{ and } d = e^{-1} \pmod{(p-1)(q-1)}.$$

$$\text{Want: } (m^e)^d = m^{ed} = m \pmod{N}.$$

Another view:

$$d = e^{-1} \pmod{(p-1)(q-1)} \iff ed = k(p-1)(q-1) + 1.$$

Consider...

Fermat's Little Theorem: For prime p , and $a \not\equiv 0 \pmod{p}$,

$$a^{p-1} \equiv 1 \pmod{p}.$$

$$\implies a^{k(p-1)} \equiv 1 \pmod{p} \implies$$

Always decode correctly?

$$E(m, (N, e)) = m^e \pmod{N}.$$

$$D(m, (N, d)) = m^d \pmod{N}.$$

$$N = pq \text{ and } d = e^{-1} \pmod{(p-1)(q-1)}.$$

$$\text{Want: } (m^e)^d = m^{ed} = m \pmod{N}.$$

Another view:

$$d = e^{-1} \pmod{(p-1)(q-1)} \iff ed = k(p-1)(q-1) + 1.$$

Consider...

Fermat's Little Theorem: For prime p , and $a \not\equiv 0 \pmod{p}$,

$$a^{p-1} \equiv 1 \pmod{p}.$$

$$\implies a^{k(p-1)} \equiv 1 \pmod{p} \implies a^{k(p-1)+1}$$

Always decode correctly?

$$E(m, (N, e)) = m^e \pmod{N}.$$

$$D(m, (N, d)) = m^d \pmod{N}.$$

$$N = pq \text{ and } d = e^{-1} \pmod{(p-1)(q-1)}.$$

$$\text{Want: } (m^e)^d = m^{ed} = m \pmod{N}.$$

Another view:

$$d = e^{-1} \pmod{(p-1)(q-1)} \iff ed = k(p-1)(q-1) + 1.$$

Consider...

Fermat's Little Theorem: For prime p , and $a \not\equiv 0 \pmod{p}$,

$$a^{p-1} \equiv 1 \pmod{p}.$$

$$\implies a^{k(p-1)} \equiv 1 \pmod{p} \implies a^{k(p-1)+1} = a \pmod{p}$$

Always decode correctly?

$$E(m, (N, e)) = m^e \pmod{N}.$$

$$D(m, (N, d)) = m^d \pmod{N}.$$

$$N = pq \text{ and } d = e^{-1} \pmod{(p-1)(q-1)}.$$

$$\text{Want: } (m^e)^d = m^{ed} = m \pmod{N}.$$

Another view:

$$d = e^{-1} \pmod{(p-1)(q-1)} \iff ed = k(p-1)(q-1) + 1.$$

Consider...

Fermat's Little Theorem: For prime p , and $a \not\equiv 0 \pmod{p}$,

$$a^{p-1} \equiv 1 \pmod{p}.$$

$$\implies a^{k(p-1)} \equiv 1 \pmod{p} \implies a^{k(p-1)+1} = a \pmod{p}$$

$$\text{versus } a^{k(p-1)(q-1)+1} = a \pmod{pq}.$$

Always decode correctly?

$$E(m, (N, e)) = m^e \pmod{N}.$$

$$D(m, (N, d)) = m^d \pmod{N}.$$

$$N = pq \text{ and } d = e^{-1} \pmod{(p-1)(q-1)}.$$

$$\text{Want: } (m^e)^d = m^{ed} = m \pmod{N}.$$

Another view:

$$d = e^{-1} \pmod{(p-1)(q-1)} \iff ed = k(p-1)(q-1) + 1.$$

Consider...

Fermat's Little Theorem: For prime p , and $a \not\equiv 0 \pmod{p}$,

$$a^{p-1} \equiv 1 \pmod{p}.$$

$$\implies a^{k(p-1)} \equiv 1 \pmod{p} \implies a^{k(p-1)+1} = a \pmod{p}$$

$$\text{versus } a^{k(p-1)(q-1)+1} = a \pmod{pq}.$$

Similar, not same, but useful.

...decoding correctness...

CRT: Isomorphism between $(a \bmod p, b \bmod q)$ and $x \pmod{pq}$

...decoding correctness...

CRT: Isomorphism between $(a \bmod p, b \bmod q)$ and $x \pmod{pq}$

$$e = d^{-1} \pmod{pq}.$$

...decoding correctness...

CRT: Isomorphism between $(a \bmod p, b \bmod q)$ and $x \pmod{pq}$

$$e = d^{-1} \pmod{pq}.$$

$$x^{ed} = x^{1+k(p-1)(q-1)} \pmod{pq}$$

...decoding correctness...

CRT: Isomorphism between $(a \bmod p, b \bmod q)$ and $x \pmod{pq}$

$$e = d^{-1} \pmod{pq}.$$

$$x^{ed} = x^{1+k(p-1)(q-1)} \pmod{pq}$$

...decoding correctness...

CRT: Isomorphism between $(a \bmod p, b \bmod q)$ and $x \pmod{pq}$

$$e = d^{-1} \pmod{pq}.$$

$$x^{ed} = x^{1+k(p-1)(q-1)} \pmod{pq}$$

Now $x = a \bmod p$ and $x = b \pmod{q}$.

...decoding correctness...

CRT: Isomorphism between $(a \bmod p, b \bmod q)$ and $x \pmod{pq}$

$$e = d^{-1} \pmod{pq}.$$

$$x^{ed} = x^{1+k(p-1)(q-1)} \pmod{pq}$$

Now $x = a \bmod p$ and $x = b \pmod{q}$.

...decoding correctness...

CRT: Isomorphism between $(a \bmod p, b \bmod q)$ and $x \pmod{pq}$

$$e = d^{-1} \pmod{pq}.$$

$$x^{ed} = x^{1+k(p-1)(q-1)} \pmod{pq}$$

Now $x = a \bmod p$ and $x = b \pmod{q}$.

$$a^{1+k(p-1)(q-1)} = a(a^{(p-1)})^{k(q-1)} = a \pmod{p}$$

...decoding correctness...

CRT: Isomorphism between $(a \bmod p, b \bmod q)$ and $x \pmod{pq}$

$$e = d^{-1} \pmod{pq}.$$

$$x^{ed} = x^{1+k(p-1)(q-1)} \pmod{pq}$$

Now $x = a \bmod p$ and $x = b \pmod{q}$.

$$a^{1+k(p-1)(q-1)} = a(a^{p-1})^{k(q-1)} = a \pmod{p}$$

$$\text{By Fermat. } a^{p-1} = 1 \pmod{p}$$

...decoding correctness...

CRT: Isomorphism between $(a \bmod p, b \bmod q)$ and $x \pmod{pq}$

$$e = d^{-1} \pmod{pq}.$$

$$x^{ed} = x^{1+k(p-1)(q-1)} \pmod{pq}$$

Now $x = a \bmod p$ and $x = b \pmod{q}$.

$$a^{1+k(p-1)(q-1)} = a(a^{(p-1)})^{k(q-1)} = a \pmod{p}$$

$$\text{By Fermat. } a^{p-1} = 1 \pmod{p}$$

$$b^{1+k(p-1)(q-1)} = b(b^{(q-1)})^{k(p-1)} = b \pmod{q}$$

...decoding correctness...

CRT: Isomorphism between $(a \bmod p, b \bmod q)$ and $x \pmod{pq}$

$$e = d^{-1} \pmod{pq}.$$

$$x^{ed} = x^{1+k(p-1)(q-1)} \pmod{pq}$$

Now $x = a \bmod p$ and $x = b \pmod{q}$.

$$a^{1+k(p-1)(q-1)} = a(a^{(p-1)})^{k(q-1)} = a \pmod{p}$$

$$\text{By Fermat. } a^{p-1} = 1 \pmod{p}$$

$$b^{1+k(p-1)(q-1)} = b(b^{(q-1)})^{k(p-1)} = b \pmod{q}$$

$$\text{By Fermat. } b^{q-1} = 1 \pmod{q}$$

...decoding correctness...

CRT: Isomorphism between $(a \bmod p, b \bmod q)$ and $x \pmod{pq}$

$$e = d^{-1} \pmod{pq}.$$

$$x^{ed} = x^{1+k(p-1)(q-1)} \pmod{pq}$$

Now $x = a \pmod{p}$ and $x = b \pmod{q}$.

$$a^{1+k(p-1)(q-1)} = a(a^{(p-1)})^{k(q-1)} = a \pmod{p}$$

$$\text{By Fermat. } a^{p-1} = 1 \pmod{p}$$

$$b^{1+k(p-1)(q-1)} = b(b^{(q-1)})^{k(p-1)} = b \pmod{q}$$

$$\text{By Fermat. } b^{q-1} = 1 \pmod{q}$$

$$x^{ed} = a \pmod{p} \text{ and } x^{ed} = b \pmod{q}.$$

...decoding correctness...

CRT: Isomorphism between $(a \bmod p, b \bmod q)$ and $x \pmod{pq}$

$$e = d^{-1} \pmod{pq}.$$

$$x^{ed} = x^{1+k(p-1)(q-1)} \pmod{pq}$$

Now $x = a \bmod p$ and $x = b \pmod{q}$.

$$a^{1+k(p-1)(q-1)} = a(a^{(p-1)})^{k(q-1)} = a \pmod{p}$$

$$\text{By Fermat. } a^{p-1} = 1 \pmod{p}$$

$$b^{1+k(p-1)(q-1)} = b(b^{(q-1)})^{k(p-1)} = b \pmod{q}$$

$$\text{By Fermat. } b^{q-1} = 1 \pmod{q}$$

$x^{ed} = a \pmod{p}$ and $x^{ed} = b \pmod{q}$.

CRT $\implies x^{ed} = x \pmod{pq}$.

Construction of keys.. ..

1. Find large (100 digit) primes p and q ?

Construction of keys.. ..

1. Find large (100 digit) primes p and q ?

Prime Number Theorem: $\pi(N)$ number of primes less than N . For all $N \geq 17$

$$\pi(N) \geq N/\ln N.$$

Construction of keys.. ..

1. Find large (100 digit) primes p and q ?

Prime Number Theorem: $\pi(N)$ number of primes less than N . For all $N \geq 17$

$$\pi(N) \geq N/\ln N.$$

Choosing randomly gives approximately $1/(\ln N)$ chance of number being a prime. (How do you tell if it is prime?)

Construction of keys.. ..

1. Find large (100 digit) primes p and q ?

Prime Number Theorem: $\pi(N)$ number of primes less than N . For all $N \geq 17$

$$\pi(N) \geq N/\ln N.$$

Choosing randomly gives approximately $1/(\ln N)$ chance of number being a prime. (How do you tell if it is prime? ...
cs170..

Construction of keys.. ..

1. Find large (100 digit) primes p and q ?

Prime Number Theorem: $\pi(N)$ number of primes less than N . For all $N \geq 17$

$$\pi(N) \geq N/\ln N.$$

Choosing randomly gives approximately $1/(\ln N)$ chance of number being a prime. (How do you tell if it is prime? ...
cs170..Miller-Rabin test..

Construction of keys.. ..

1. Find large (100 digit) primes p and q ?

Prime Number Theorem: $\pi(N)$ number of primes less than N . For all $N \geq 17$

$$\pi(N) \geq N/\ln N.$$

Choosing randomly gives approximately $1/(\ln N)$ chance of number being a prime. (How do you tell if it is prime? ... cs170..Miller-Rabin test.. Primes in P).

Construction of keys.. ..

1. Find large (100 digit) primes p and q ?

Prime Number Theorem: $\pi(N)$ number of primes less than N . For all $N \geq 17$

$$\pi(N) \geq N/\ln N.$$

Choosing randomly gives approximately $1/(\ln N)$ chance of number being a prime. (How do you tell if it is prime? ... cs170..Miller-Rabin test.. Primes in P).

For 1024 bit number, 1 in 710 is prime.

Construction of keys.. ..

1. Find large (100 digit) primes p and q ?

Prime Number Theorem: $\pi(N)$ number of primes less than N . For all $N \geq 17$

$$\pi(N) \geq N/\ln N.$$

Choosing randomly gives approximately $1/(\ln N)$ chance of number being a prime. (How do you tell if it is prime? ... cs170..Miller-Rabin test.. Primes in P).

For 1024 bit number, 1 in 710 is prime.

2. Choose e with $\gcd(e, (p-1)(q-1)) = 1$.

Construction of keys.. ..

1. Find large (100 digit) primes p and q ?

Prime Number Theorem: $\pi(N)$ number of primes less than N . For all $N \geq 17$

$$\pi(N) \geq N/\ln N.$$

Choosing randomly gives approximately $1/(\ln N)$ chance of number being a prime. (How do you tell if it is prime? ... cs170..Miller-Rabin test.. Primes in P).

For 1024 bit number, 1 in 710 is prime.

2. Choose e with $\gcd(e, (p-1)(q-1)) = 1$.
Use gcd algorithm to test.

Construction of keys.. ..

1. Find large (100 digit) primes p and q ?

Prime Number Theorem: $\pi(N)$ number of primes less than N . For all $N \geq 17$

$$\pi(N) \geq N/\ln N.$$

Choosing randomly gives approximately $1/(\ln N)$ chance of number being a prime. (How do you tell if it is prime? ... cs170..Miller-Rabin test.. Primes in P).

For 1024 bit number, 1 in 710 is prime.

2. Choose e with $\gcd(e, (p-1)(q-1)) = 1$.
Use gcd algorithm to test.
3. Find inverse d of e modulo $(p-1)(q-1)$.

Construction of keys.. ..

1. Find large (100 digit) primes p and q ?

Prime Number Theorem: $\pi(N)$ number of primes less than N . For all $N \geq 17$

$$\pi(N) \geq N/\ln N.$$

Choosing randomly gives approximately $1/(\ln N)$ chance of number being a prime. (How do you tell if it is prime? ... cs170..Miller-Rabin test.. Primes in P).

For 1024 bit number, 1 in 710 is prime.

2. Choose e with $\gcd(e, (p-1)(q-1)) = 1$.
Use gcd algorithm to test.
3. Find inverse d of e modulo $(p-1)(q-1)$.
Use extended gcd algorithm.

Construction of keys.. ..

1. Find large (100 digit) primes p and q ?

Prime Number Theorem: $\pi(N)$ number of primes less than N . For all $N \geq 17$

$$\pi(N) \geq N/\ln N.$$

Choosing randomly gives approximately $1/(\ln N)$ chance of number being a prime. (How do you tell if it is prime? ... cs170..Miller-Rabin test.. Primes in P).

For 1024 bit number, 1 in 710 is prime.

2. Choose e with $\gcd(e, (p-1)(q-1)) = 1$.
Use gcd algorithm to test.
3. Find inverse d of e modulo $(p-1)(q-1)$.
Use extended gcd algorithm.

All steps are polynomial in $O(\log N)$, the number of bits.

Security of RSA.

Security of RSA.

Security?

1. Alice knows p and q .
2. Bob only knows, $N(= pq)$, and e .

Security of RSA.

Security?

1. Alice knows p and q .
2. Bob only knows, $N(= pq)$, and e .

Does not know, for example, d or factorization of N .

Security of RSA.

Security?

1. Alice knows p and q .
2. Bob only knows, $N(= pq)$, and e .
Does not know, for example, d or factorization of N .
3. I don't know how to break this scheme without factoring N .

Security of RSA.

Security?

1. Alice knows p and q .
2. Bob only knows, $N(= pq)$, and e .
Does not know, for example, d or factorization of N .
3. I don't know how to break this scheme without factoring N .

No one I know or have heard of admits to knowing how to factor N .

Security of RSA.

Security?

1. Alice knows p and q .
2. Bob only knows, $N(= pq)$, and e .
Does not know, for example, d or factorization of N .
3. I don't know how to break this scheme without factoring N .

No one I know or have heard of admits to knowing how to factor N .
Breaking in general sense \implies factoring algorithm.

Much more to it.....

If Bobs sends a message (Credit Card Number) to Alice,

Much more to it.....

If Bobs sends a message (Credit Card Number) to Alice,
Eve sees it.

Much more to it.....

If Bobs sends a message (Credit Card Number) to Alice,
Eve sees it.

Eve can send credit card again!!

Much more to it.....

If Bobs sends a message (Credit Card Number) to Alice,
Eve sees it.

Eve can send credit card again!!

The protocols are built on RSA but more complicated;

Much more to it.....

If Bobs sends a message (Credit Card Number) to Alice,
Eve sees it.

Eve can send credit card again!!

The protocols are built on RSA but more complicated;
For example, several rounds of challenge/response.

Much more to it.....

If Bob sends a message (Credit Card Number) to Alice,
Eve sees it.

Eve can send credit card again!!

The protocols are built on RSA but more complicated;
For example, several rounds of challenge/response.

One trick:

Bob encodes credit card number, c ,

Much more to it.....

If Bob sends a message (Credit Card Number) to Alice,
Eve sees it.

Eve can send credit card again!!

The protocols are built on RSA but more complicated;
For example, several rounds of challenge/response.

One trick:

Bob encodes credit card number, c ,
concatenated with random k -bit number r .

Much more to it.....

If Bob sends a message (Credit Card Number) to Alice,
Eve sees it.

Eve can send credit card again!!

The protocols are built on RSA but more complicated;
For example, several rounds of challenge/response.

One trick:

Bob encodes credit card number, c ,
concatenated with random k -bit number r .

Never sends just c .

Much more to it.....

If Bob sends a message (Credit Card Number) to Alice,
Eve sees it.

Eve can send credit card again!!

The protocols are built on RSA but more complicated;
For example, several rounds of challenge/response.

One trick:

Bob encodes credit card number, c ,
concatenated with random k -bit number r .

Never sends just c .

Again, more work to do to get entire system.

Much more to it.....

If Bob sends a message (Credit Card Number) to Alice,
Eve sees it.

Eve can send credit card again!!

The protocols are built on RSA but more complicated;
For example, several rounds of challenge/response.

One trick:

Bob encodes credit card number, c ,
concatenated with random k -bit number r .

Never sends just c .

Again, more work to do to get entire system.

CS161...

Signatures using RSA.

Verisign:



Signatures using RSA.

Verisign:

Amazon

Browser.



Certificate Authority: Verisign, GoDaddy, DigiNotar,...

Signatures using RSA.

Verisign: k_v , K_v

Amazon

Browser.



Certificate Authority: Verisign, GoDaddy, DigiNotar,...

Verisign's key: $K_v = (N, e)$ and $k_v = d$ ($N = pq$.)

Signatures using RSA.

Verisign: k_V , K_V

Amazon

Browser. K_V



Certificate Authority: Verisign, GoDaddy, DigiNotar,...

Verisign's key: $K_V = (N, e)$ and $k_V = d$ ($N = pq$.)

Browser "knows" Verisign's public key: K_V .

Signatures using RSA.

Verisign: k_v , K_v

Amazon

Browser. K_v

Certificate Authority: Verisign, GoDaddy, DigiNotar,...

Verisign's key: $K_v = (N, e)$ and $k_v = d$ ($N = pq$.)

Browser "knows" Verisign's public key: K_v .

Amazon Certificate: C = "I am Amazon. My public Key is K_A ."

Signatures using RSA.

Verisign: k_V, K_V

$[C, S_V(C)]$

Amazon

Browser. K_V

Certificate Authority: Verisign, GoDaddy, DigiNotar,...

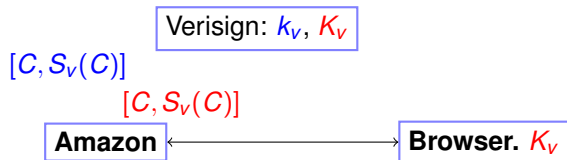
Verisign's key: $K_V = (N, e)$ and $k_V = d$ ($N = pq$.)

Browser "knows" Verisign's public key: K_V .

Amazon Certificate: $C =$ "I am Amazon. My public Key is K_A ."

Verisign signature of C : $S_V(C)$: $D(C, k_V) = C^d \pmod N$.

Signatures using RSA.



Certificate Authority: Verisign, GoDaddy, DigiNotar,...

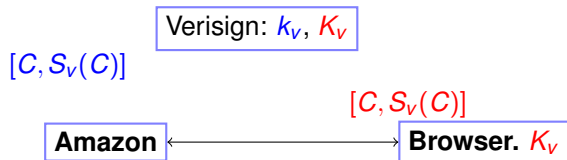
Verisign's key: $K_V = (N, e)$ and $k_V = d$ ($N = pq$.)

Browser "knows" Verisign's public key: K_V .

Amazon Certificate: $C =$ "I am Amazon. My public Key is K_A ."

Verisign signature of C : $S_V(C)$: $D(C, k_V) = C^d \pmod N$.

Signatures using RSA.



Certificate Authority: Verisign, GoDaddy, DigiNotar,...

Verisign's key: $K_V = (N, e)$ and $k_V = d$ ($N = pq$.)

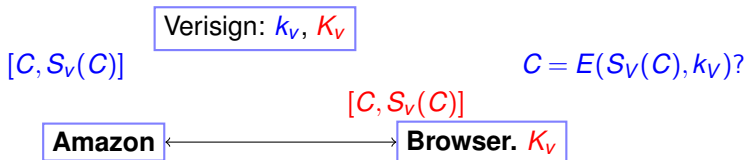
Browser "knows" Verisign's public key: K_V .

Amazon Certificate: $C =$ "I am Amazon. My public Key is K_A ."

Verisign signature of C : $S_V(C)$: $D(C, k_V) = C^d \pmod N$.

Browser receives: $[C, y]$

Signatures using RSA.



Certificate Authority: Verisign, GoDaddy, DigiNotar,...

Verisign's key: $K_V = (N, e)$ and $k_V = d$ ($N = pq$.)

Browser "knows" Verisign's public key: K_V .

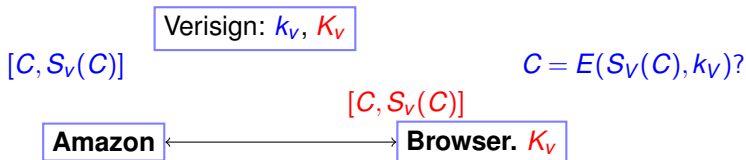
Amazon Certificate: $C =$ "I am Amazon. My public Key is K_A ."

Verisign signature of C : $S_V(C)$: $D(C, k_V) = C^d \pmod N$.

Browser receives: $[C, y]$

Checks $E(y, K_V) = C?$

Signatures using RSA.



Certificate Authority: Verisign, GoDaddy, DigiNotar,...

Verisign's key: $K_V = (N, e)$ and $k_V = d$ ($N = pq$.)

Browser "knows" Verisign's public key: K_V .

Amazon Certificate: $C =$ "I am Amazon. My public Key is K_A ."

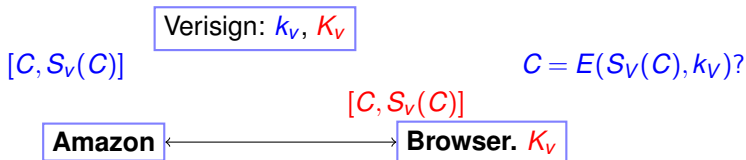
Verisign signature of C : $S_V(C)$: $D(C, k_V) = C^d \pmod N$.

Browser receives: $[C, y]$

Checks $E(y, K_V) = C?$

$E(S_V(C), K_V)$

Signatures using RSA.



Certificate Authority: Verisign, GoDaddy, DigiNotar,...

Verisign's key: $K_V = (N, e)$ and $k_V = d$ ($N = pq$.)

Browser "knows" Verisign's public key: K_V .

Amazon Certificate: $C =$ "I am Amazon. My public Key is K_A ."

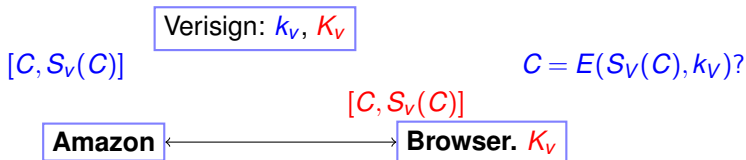
Verisign signature of C : $S_V(C)$: $D(C, k_V) = C^d \pmod N$.

Browser receives: $[C, y]$

Checks $E(y, K_V) = C?$

$E(S_V(C), K_V) = (S_V(C))^e$

Signatures using RSA.



Certificate Authority: Verisign, GoDaddy, DigiNotar,...

Verisign's key: $K_V = (N, e)$ and $k_V = d$ ($N = pq$.)

Browser "knows" Verisign's public key: K_V .

Amazon Certificate: $C =$ "I am Amazon. My public Key is K_A ."

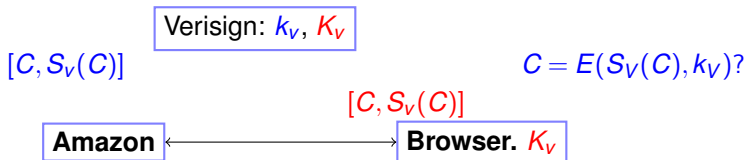
Verisign signature of C : $S_V(C)$: $D(C, k_V) = C^d \pmod N$.

Browser receives: $[C, y]$

Checks $E(y, K_V) = C?$

$$E(S_V(C), K_V) = (S_V(C))^e = (C^d)^e$$

Signatures using RSA.



Certificate Authority: Verisign, GoDaddy, DigiNotar,...

Verisign's key: $K_V = (N, e)$ and $k_V = d$ ($N = pq$.)

Browser "knows" Verisign's public key: K_V .

Amazon Certificate: $C =$ "I am Amazon. My public Key is K_A ."

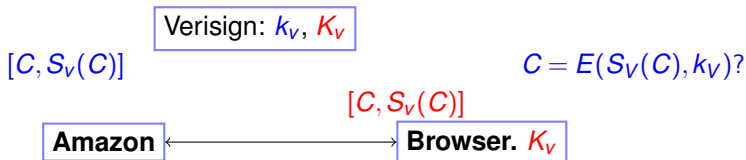
Verisign signature of C : $S_V(C)$: $D(C, k_V) = C^d \pmod N$.

Browser receives: $[C, y]$

Checks $E(y, K_V) = C?$

$$E(S_V(C), K_V) = (S_V(C))^e = (C^d)^e = C^{de}$$

Signatures using RSA.



Certificate Authority: Verisign, GoDaddy, DigiNotar,...

Verisign's key: $K_V = (N, e)$ and $k_V = d$ ($N = pq$.)

Browser "knows" Verisign's public key: K_V .

Amazon Certificate: $C =$ "I am Amazon. My public Key is K_A ."

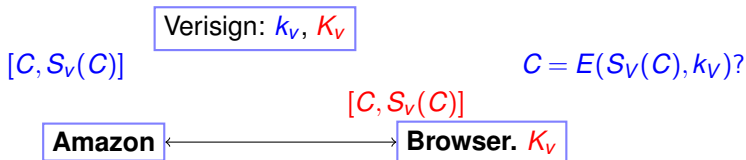
Verisign signature of C : $S_V(C)$: $D(C, k_V) = C^d \pmod N$.

Browser receives: $[C, y]$

Checks $E(y, K_V) = C?$

$E(S_V(C), K_V) = (S_V(C))^e = (C^d)^e = C^{de} = C \pmod N$

Signatures using RSA.



Certificate Authority: Verisign, GoDaddy, DigiNotar,...

Verisign's key: $K_V = (N, e)$ and $k_V = d$ ($N = pq$.)

Browser "knows" Verisign's public key: K_V .

Amazon Certificate: $C =$ "I am Amazon. My public Key is K_A ."

Verisign signature of C : $S_V(C)$: $D(C, k_V) = C^d \pmod N$.

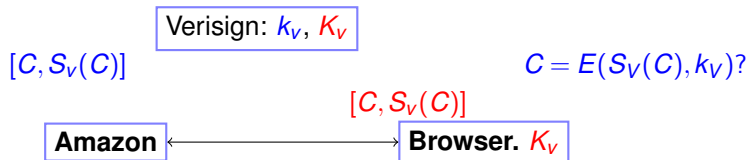
Browser receives: $[C, y]$

Checks $E(y, K_V) = C?$

$E(S_V(C), K_V) = (S_V(C))^e = (C^d)^e = C^{de} = C \pmod N$

Valid signature of Amazon certificate $C!$

Signatures using RSA.



Certificate Authority: Verisign, GoDaddy, DigiNotar,...

Verisign's key: $K_V = (N, e)$ and $k_V = d$ ($N = pq$.)

Browser "knows" Verisign's public key: K_V .

Amazon Certificate: $C =$ "I am Amazon. My public Key is K_A ."

Verisign signature of C : $S_V(C)$: $D(C, k_V) = C^d \pmod N$.

Browser receives: $[C, y]$

Checks $E(y, K_V) = C?$

$E(S_V(C), K_V) = (S_V(C))^e = (C^d)^e = C^{de} = C \pmod N$

Valid signature of Amazon certificate $C!$

Security: Eve can't forge unless she "breaks" RSA scheme.

RSA

Public Key Cryptography:

RSA

Public Key Cryptography:

$$D(E(m, K), k) = (m^e)^d \bmod N = m.$$

RSA

Public Key Cryptography:

$$D(E(m, K), k) = (m^e)^d \bmod N = m.$$

Signature scheme:

RSA

Public Key Cryptography:

$$D(E(m, K), k) = (m^e)^d \bmod N = m.$$

Signature scheme:

$$E(D(C, k), K) = (C^d)^e \bmod N = C$$

Other Eve.

Other Eve.

Get CA to certify fake certificates: Microsoft Corporation.

Other Eve.

Get CA to certify fake certificates: Microsoft Corporation.
2001..Doh.

Other Eve.

Get CA to certify fake certificates: Microsoft Corporation.

2001..Doh.

... and August 28, 2011 announcement.

Other Eve.

Get CA to certify fake certificates: Microsoft Corporation.

2001..Doh.

... and August 28, 2011 announcement.

DigiNotar Certificate issued for Microsoft!!!

Other Eve.

Get CA to certify fake certificates: Microsoft Corporation.

2001..Doh.

... and August 28, 2011 announcement.

DigiNotar Certificate issued for Microsoft!!!

How does Microsoft get a CA to issue certificate to them ...

Other Eve.

Get CA to certify fake certificates: Microsoft Corporation.

2001..Doh.

... and August 28, 2011 announcement.

DigiNotar Certificate issued for Microsoft!!!

How does Microsoft get a CA to issue certificate to them ...

and only them?

Summary.

Public-Key Encryption.

Summary.

Public-Key Encryption.

RSA Scheme:

Summary.

Public-Key Encryption.

RSA Scheme:

$$N = pq \text{ and } d = e^{-1} \pmod{(p-1)(q-1)}.$$

$$E(x) = x^e \pmod{N}.$$

$$D(y) = y^d \pmod{N}.$$

Summary.

Public-Key Encryption.

RSA Scheme:

$N = pq$ and $d = e^{-1} \pmod{(p-1)(q-1)}$.

$$E(x) = x^e \pmod{N}.$$

$$D(y) = y^d \pmod{N}.$$

Repeated Squaring \implies efficiency.

Summary.

Public-Key Encryption.

RSA Scheme:

$N = pq$ and $d = e^{-1} \pmod{(p-1)(q-1)}$.

$E(x) = x^e \pmod{N}$.

$D(y) = y^d \pmod{N}$.

Repeated Squaring \implies efficiency.

Fermat's Theorem \implies correctness.

Summary.

Public-Key Encryption.

RSA Scheme:

$N = pq$ and $d = e^{-1} \pmod{(p-1)(q-1)}$.

$$E(x) = x^e \pmod{N}.$$

$$D(y) = y^d \pmod{N}.$$

Repeated Squaring \implies efficiency.

Fermat's Theorem \implies correctness.

Good for Encryption

Summary.

Public-Key Encryption.

RSA Scheme:

$N = pq$ and $d = e^{-1} \pmod{(p-1)(q-1)}$.

$E(x) = x^e \pmod{N}$.

$D(y) = y^d \pmod{N}$.

Repeated Squaring \implies efficiency.

Fermat's Theorem \implies correctness.

Good for Encryption and Signature Schemes.