CS 4173/5173
COMPUTER SECURITY

High Level Introduction to Public Key Cryptography

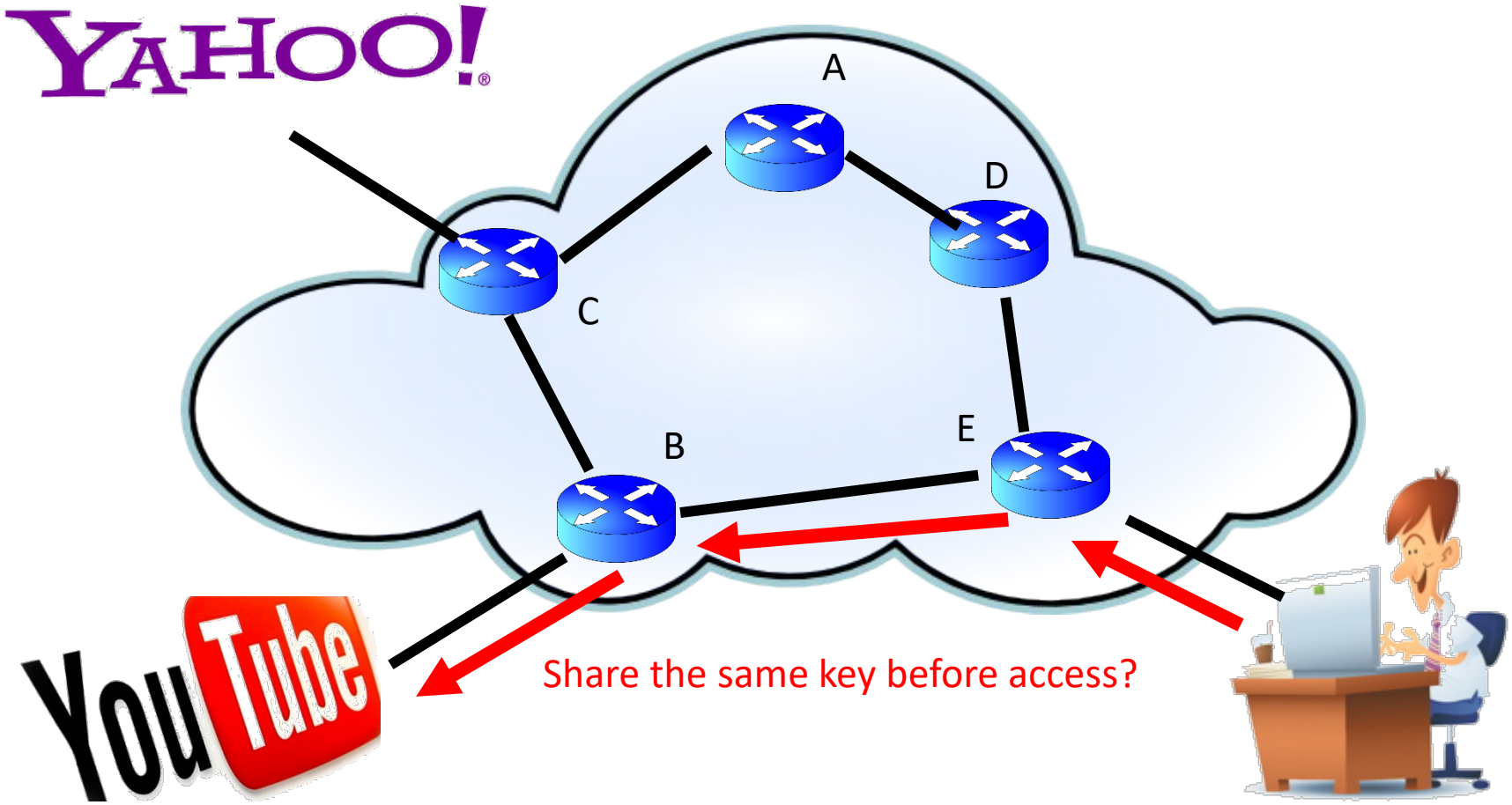
COMPANY A'S PROBLEM I

- Company A is a big web service company with over 10,000 employees.
- The president Bob wants to make sure that all employees can verify the authenticity of the announcement emails that he sends.
- Q: How to ensure authenticity of these emails.

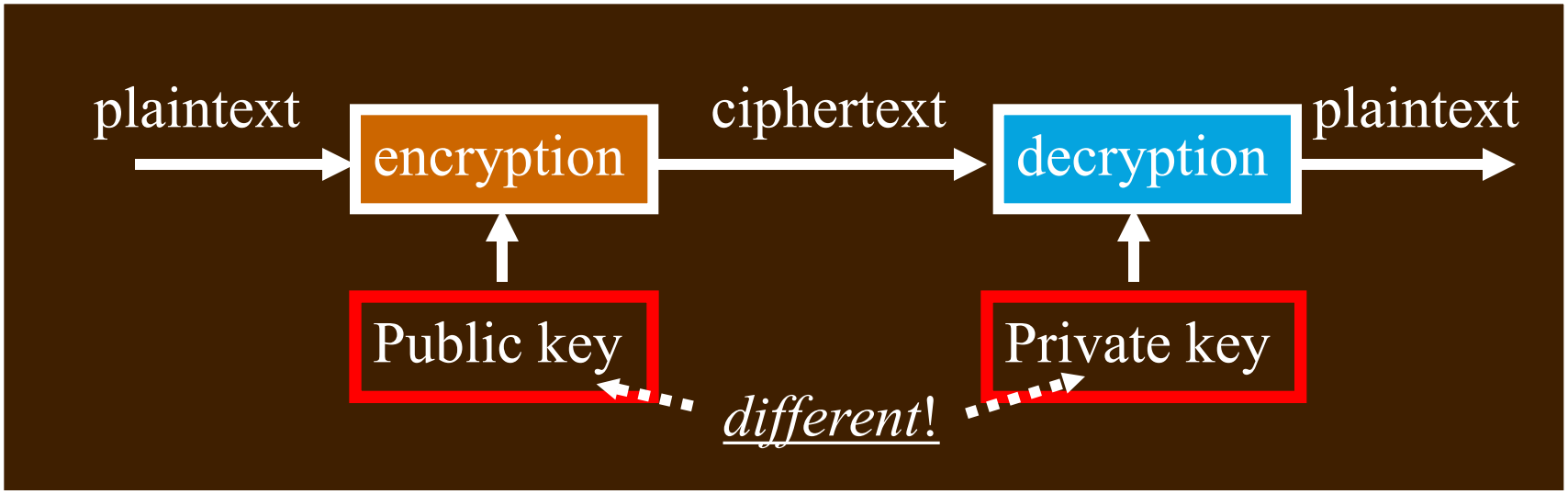
COMPANY A'S PROBLEM II

- Company A is accepting vulnerability report of their web system from the public.
- They need a design that someone can successfully send the report of a potential vulnerability via email to them.
- Q: How to ensure the confidentiality of reports?

HOW TO SECURELY SURF INTERNET



PUBLIC KEY CRYPTOGRAPHY

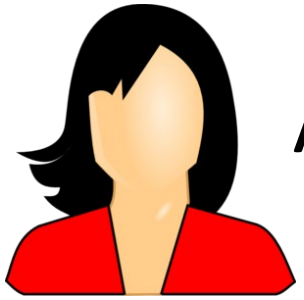


- Invented and published in 1975
- A *public / private key pair* is used
- Also known as *asymmetric* cryptography
- Much slower to compute than secret key cryptography

PUBLIC/PRIVATE KEY

- Public key – encrypt
- Private key – decrypt

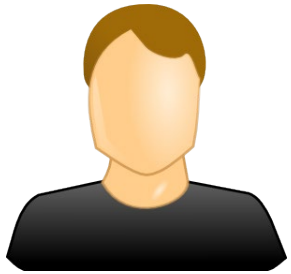
- How does the secret communication look like?



Alice

wants to send a message

Bob



PUBLIC/PRIVATE KEY

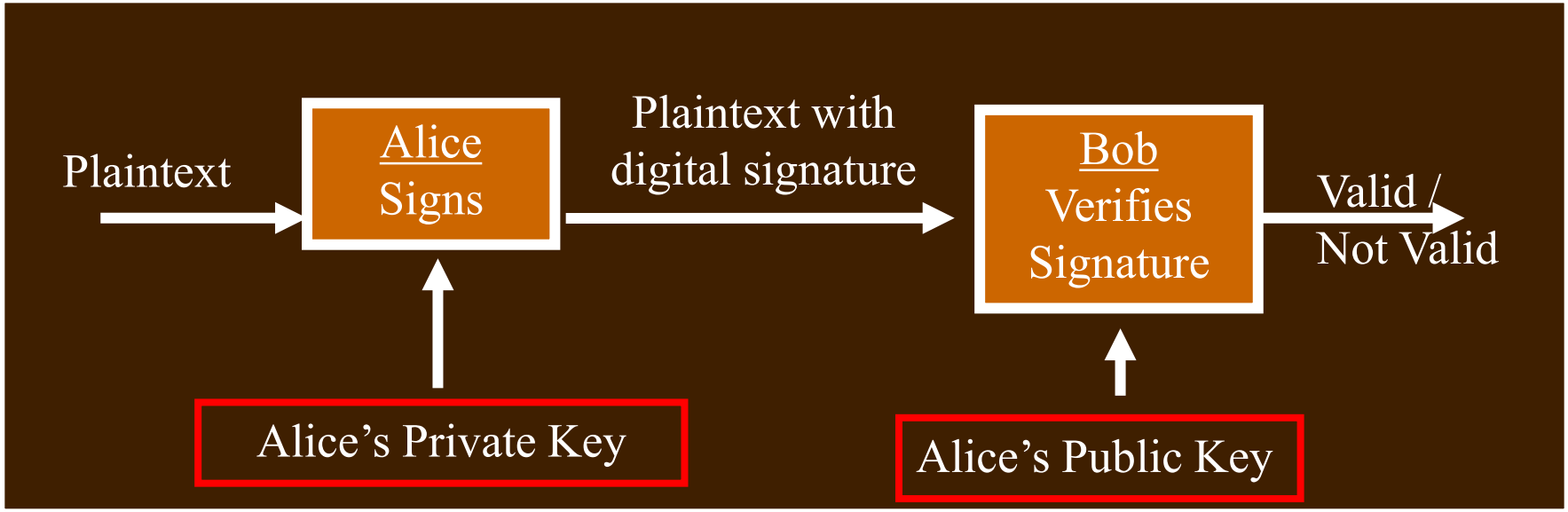
- Alice has her own public and private key pair
- Bob also has his own public and private key pair

- Public key
  - Can be released to the public

- Private Key
  - Must be kept secret.

AUTHENTICATION IN PUBLIC KEY CRYPTO

- Message integrity with digital signatures
- Alice computes hash, signs with her private key (no one else can do this without her key)
- Bob verifies hash on receipt using Alice's public key using the verification equation



AUTHENTICATION (CONT'D)

- Authentication in public key crypto:
  - Hash function to hash the message into a digest
  - The action of sign the digest with (private key)
  - The action of verify the digest with (public key)

PUBLIC-KEY REQUIREMENTS

- It must be computationally
  - easy to generate a public / private key pair
  - hard to determine the private key, given the public key
- It must be computationally
  - easy to encrypt using the public key
  - easy to decrypt using the private key
  - hard to recover the plaintext message from just the ciphertext and the public key

PUBLIC KEY ALGORITHMS

- Public key algorithms covered in this class, and their applications

System	Encryption / Decryption?	Digital Signatures?	Key Exchange?
RSA	Yes	Yes	Yes
Diffie-Hellman			Yes
DSA		Yes	

SOLVING COMPANY A'S PROBLEM I

- Company A is a big web service company with over 10,000 employees.
- The president Bob wants to make sure that all employees can verify the authenticity of the announcement emails that he sends.
- Answer:
  - Everyone knows Bob's public key.
  - Bob signs the email using his private key.
  - Everyone can verify the signed email using Bob's public key

SOLVING COMPANY A'S PROBLEM II

- Company A is accepting vulnerability report of their web system from the public.
- They need a design that someone can successfully send the report of a potential vulnerability via email to them.
- Answer:
  - Company A generates a key pair, then releases the public key to the public for vulnerability report.
  - Everyone uses the public key to encrypt the report.

PUBLIC KEY VS. SYMMETRIC KEY

Symmetric key	Public key
Two parties MUST trust each other	Two parties DO NOT need to trust each other
Both share same key	Two separate keys: a public and a private key
Typically faster	Typically slower
Examples: DES, RC5, AES, …	Examples: RSA, DSA, ECC…

COMPANY A'S PROBLEM III

- Company A provides an on-line chat service for vulnerability report.
  - Requirement 1: confidentiality.
  - Requirement 2: efficiency because there will be a number of message exchanges.

- Q: How to satisfy both requirements?

DIGITAL ENVELOPE: SYMMETRIC+ASYMMETRIC

1. Generate a secret key (called a session key) at random.
2. Encrypt the message using the session key and symmetric algorithm.
3. Encrypt the session key with the recipient's public key. This becomes the "digital envelope".
4. Send the encrypted message and the digital envelope to the recipient.

DIGITAL ENVELOPE (CONT'D)

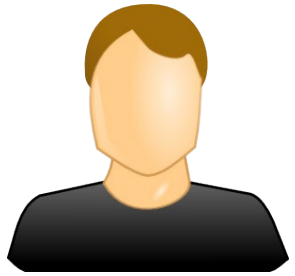
Alice (finds a vulnerability)

Bob (company representative)

Knows the public key: Kp

Generate a session key Ks

Encrypt message M using Ks
Encrypted Ks using Kp



Public key: Kp
Private key: Ki

1. Use Ki to decrypt Ks
2. Use Ks to decrypt M

Send a response using Ks

All remaining communication is based on Ks

CS 4173/5173 COMPUTER SECURITY

Basic Number Theory I

Divisors, GCD and Euclid's Algorithm

SOME REVIEW: DIVISORS

- Set of all integers is $\mathcal{Z} = \{\ldots,-2,-1,0,1,2,\ldots\}$
- *b divides a* (or *b* is a *divisor* of *a*) if $a = mb$ for some integer *m*
  - Denoted as b | a
  - any $b \neq 0$ divides 0
- For any *a*, 1 and *a* are *trivial divisors* of *a*
  - all other divisors of *a* are called *factors* of *a*
- Q: Is 2 a factor of 8?

REMAINDERS AND CONGRUENCY

- For any integer a and any positive integer n, there are two unique integers q and r, such that $0 \leq r < n$ and $a = qn + r$
  - r is the *remainder* of a divided by n, written $r = a \bmod n$

Example: $12 = 2*5 + 2$ ➔ $2 = 12 \bmod 5$

- a and b are *congruent* modulo n, written **$a \equiv b \bmod n$**, if $a \bmod n = b \bmod n$

Example: $7 \bmod 5 = 12 \bmod 5$ ➔ $7 \equiv 12 \bmod 5$

REMAINDERS (CONT'D)

- For any positive integer *n*, the integers can be divided into *n* equivalence classes according to their remainders modulo *n*
  - denote the set as $\mathcal{Z}_n$
- i.e., the (mod *n*) operator maps all integers into the set of integers
  - $\mathcal{Z}_n$={0, 1, 2, …, (*n*-1)}

EXERCISES

- 100 mod 10 = ?
- 100 mod 11 = ?

- True or False:
  - 2 and 5 are congruent mod 2
  - 3 and 5 are congruent mod 2
  - 4 and 5 are congruent mod 2
  - 10 and 5 are congruent mod 2

PRIMES AND FACTORS

- *a* is *prime* if it has no non-trivial factors
  - examples: 2, 3, 5, 7, 11, 13, 17, 19, 31,…
- Theorem: there are infinitely many primes
- (Factorization) Any integer $a > 1$ can be factored in a unique way as $p_1^{a_1} \bullet p_2^{a_2} \bullet \ldots p_t^{a_t}$
  - where all $p_1 > p_2 > \ldots > p_t$ are prime numbers and where each $a_i > 0$

Examples:

$91 = 13^1 \times 7^1$

$11{,}011 = 13^1 \times 11^2 \times 7^1$

COMMON DIVISORS

- A number *d* that is a divisor of both *a* and *b* is a *common divisor* of *a* and *b*

Example: common divisors of 30 and 24 are 1, 2, 3, 6

- If $d|a$ and $d|b$, then $d|(a+b)$ and $d|(a-b)$

Example: Since 3 | 30 and 3 | 24 , 3 | (30+24) and 3 | (30-24)

- If $d|a$ and $d|b$, then $d|(ax+by)$ for any integers *x* and *y*

Example: 3 | 30 and 3 | 24 ➔ 3 | (2*30 + 6*24)

EXERCISE

- What are the common divisors of 100 and 20?

GREATEST COMMON DIVISOR (GCD)

- *gcd(a,b)* = max{*k* | *k*|*a* and *k*|*b*}

Example: gcd(60,24) = 12, gcd(a,0) = a

- Properties:
  - if 0 ≤ *n*, then gcd(*an*, *bn*) = *n**gcd(*a*,*b*)
  - gcd(a,0) = a
  - If gcd(*a*,*b*)=1, *a* and *b* are relatively prime.
  - For all positive integers *d*, *a*, and *b*, if *d* | *ab* and gcd(*a*,*d*) = 1
    - then *d*|*b*
  - Example:
    - 3 | 4*9, gcd(3, 4) = 1, ➔ 3 | 9

GCD (CONT'D)

- Computing GCD by hand:
  - $a = p_1^{a1} p_2^{a2} \ldots p_r^{ar}$
    $b = p_1^{b1} p_2^{b2} \ldots p_r^{br}$
    - where $p_1 < p_2 < \ldots < p_r$ are prime,
    - and *ai* and *bi* are nonnegative,
    - then $\gcd(a, b) = p_1^{\min(a1, b1)} p_2^{\min(a2, b2)} \ldots p_r^{\min(ar, br)}$

$\Rightarrow$Slow way to find the GCD

- requires factoring *a* and *b* first

EUCLID'S ALGORITHM FOR GCD

- Insight:
  gcd(*x, y*) = gcd(*y, x* mod *y*)
- Procedure **`euclid(x, y)`** :

```
r[0] = x, r[1] = y, n = 1;
while (r[n] != 0) {
    n = n+1;
    r[n] = r[n-2] % r[n-1];
}
return r[n-1];
```

EXAMPLE

X=595, y=408

n	*r(n)*
0	595
1	408
2	595 mod 408 = 187
3	408 mod 187 = 34
4	187 mod 34 = 17
5	34 mod 17 = 0

gcd(595,408) = 17

```
r[0] = x, r[1] = y, n = 1;
while (r[n] != 0) {
    n = n+1;
    r[n] = r[n-2] % r[n-1];
}
return r[n-1];
```

EXERCISE

X=120, y=1000

n	*r(n)*
0	120
1	1000
2	120 mod 1000 = 120
3	1000 mod 120 = 40
4	120 mod 40 = 0
5	

gcd(120,1000) = 40

```
r[0] = x, r[1] = y, n = 1;
while (r[n] != 0) {
    n = n+1;
    r[n] = r[n-2] % r[n-1];
}
return r[n-1];
```

EXTENDED EUCLID'S ALGORITHM

- Let $\mathcal{LC}(x,y) = \{ux+vy : x,y \in \mathcal{Z}\}$ be the set of linear combinations of *x* and *y*
  - u and v are intergers (can be negative).
- Theorem: if *x* and *y* are any integers > 0, then gcd(*x*,*y*) is the smallest positive element of $\mathcal{LC}(x,y)$
- Euclid's algorithm can be extended to compute *u* and *v*, as well as gcd(*x*,*y*)

EXTENDED EUCLID'S ALGORITHM

```
r[0] = x, r[1] = y, n = 1;
u[0] = 1, u[1] = 0;
v[0] = 0, v[1] = 1;
while (r[n] != 0) {
  n = n+1;
  r[n] = r[n-2] % r[n-1];
  q[n] = (int) (r[n-2] / r[n-1]);
  u[n] = u[n-2] - q[n]*u[n-1];
  v[n] = v[n-2] - q[n]*v[n-1];
}
return r[n-1], u[n-1], v[n-1];
```

C-style floor function

EXTENDED EUCLID'S EXAMPLE

n	q_n	r_n	u_n	v_n
0	-	595	1	0
1	-	408	0	1
2	1	187	1	-1
3	2	34	-2	3
4	5	17	11	-16
5	2	0	-24	35

gcd(595,408) = 17 = 11*595 + -16*408

CS 4173/5173
COMPUTER SECURITY

Review of Modular Arithmetic

MODULAR ARITHMETIC

- Modular addition
  - $[(a \bmod n) + (b \bmod n)] \bmod n = (a+b) \bmod n$

Example: $[16 \bmod 12 + 8 \bmod 12] \bmod 12 = (16 + 8) \bmod 12 = 0$

- Modular subtraction
  - $[(a \bmod n) - (b \bmod n)] \bmod n = (a - b) \bmod n$

Example: $[22 \bmod 12 - 8 \bmod 12] \bmod 12 = (22 - 8) \bmod 12 = 2$

- Modular multiplication
  - $[(a \bmod n) \times (b \bmod n)] \bmod n = (a \times b) \bmod n$

Example: $[22 \bmod 12 \times 8 \bmod 12] \bmod 12 = (22 \times 8) \bmod 12 = 8$

EXERCISE

- (199 mod 10 + 111 mod 10) mod 10 = ?
- ((192939 mod 11) * (22 mod 11)) mod 11 = ?

PROPERTIES OF MODULAR ARITHMETIC

- Commutative laws
  - $(w + x) \bmod n = (x + w) \bmod n$
  - $(w \times x) \bmod n = (x \times w) \bmod n$
- Associative laws
  - $[(w + x) + y] \bmod n = [w + (x + y)] \bmod n$
  - $[(w \times x) \times y] \bmod n = [w \times (x \times y)] \bmod n$
- Distributive law
  - $[w \times (x + y)] \bmod n = [(w \times x)+(w \times y)] \bmod n$

PROPERTIES (CONT'D)

- Idempotent elements
  - $(0 + m) \bmod n = m \bmod n$
  - $(1 \times m) \bmod n = m \bmod n$
- Additive inverse ($-w$)
  - for each $m \in \mathcal{Z}_n$, there exists z such that $(m + z) \bmod n = 0$

Example: 3 and 4 are additive inverses mod 7, since $(3 + 4) \bmod 7 = 0$

- Multiplicative inverse
  - for each positive $m \in \mathcal{Z}_n$, is there a z such that
    - (m * z) mod n = 1

EXERCISE

- Are the following additive inverses mod 7 ?
  - 1 and 2
  - 3 and 11
  - 22 and 11

- Are the following multiplicative inverses mod 7?
  - 1 and 2
  - 3 and 4
  - 3 and 5
  - 3 and 11

MULTIPLICATIVE INVERSES

- Don't always exist!
  - Ex.: there is no *z* such that $6 \times z = 1 \bmod 8$ (m = 6 and n = 8)

z	0	1	2	3	4	5	6	7
6×z	0	6	12	18	24	30	36	42
6×z mod 8	0	6	4	2	0	6	4	2

- A positive integer $m \in \mathbb{Z}_n$ has a multiplicative inverse $m^{-1} \bmod n$ if and only if (iff) gcd(*m, n*) = 1, i.e., *m* and *n* are relatively prime

  $\Rightarrow$ If *n* is a prime number, then all positive elements in $\mathbb{Z}_n$ have multiplicative inverses

INVERSES (CONT'D)

- Question: does m=5 have a multiplicative inverse mod n=8?
  - Test if gcd(m, n) = 1.

z	0	1	2	3	4	5	6	7
5×z	0	5	10	15	20	25	30	35
5×z mod 8	0	5	2	7	4	1	6	3

FINDING THE MULTIPLICATIVE INVERSE

- Given m and n, how do you find m^{-1} mod n?
- Extended Euclid's Algorithm **`exteuclid(m,n)`**: m^{-1} mod n = **`v`**$_{x-1}$
  - if gcd(m,n) ≠ 1 there is no multiplicative inverse m^{-1} mod n

EXAMPLE

- If m = 12, find m^{-1} mod 35
  - Q1: does the inverse exist?
  - Q2: how to find it

EXAMPLE

m = 12, find m^{-1} mod 35

➔ Compute gcd(35, 12)

x	q_x	r_x	u_x	v_x
0	-	35	1	0
1	-	12	0	1
2	2	11	1	-2
3	1	1	-1	3
4	11	0	12	-35

```
r[0] = x, r[1] = y, n = 1;
u[0] = 1, u[1] = 0;
v[0] = 0, v[1] = 1;
while (r[n] != 0) {
  n = n+1;
  r[n] = r[n-2] % r[n-1];
  q[n] = (int) (r[n-2] / r[n-1]);
  u[n] = u[n-2] - q[n]*u[n-1];
  v[n] = v[n-2] - q[n]*v[n-1];
}
return r[n-1], u[n-1], v[n-1];
```

m^{-1} mod n = $\mathbf{v_{n-1}}$

MODULAR DIVISION

- If the multiplicative inverse of b mod n exists, then $(a \bmod n) / (b \bmod n) = (a * (b^{-1} \bmod n)) \bmod n$

Example: $(13 \bmod 11) / (4 \bmod 11) = (13*(4^{-1} \bmod 11)) \bmod 11 = (13 * 3) \bmod 11 = 6$

Example: $(8 \bmod 10) / (4 \bmod 10)$ not well defined since 4 does not have a multiplicative inverse mod 10

EXERCISE

- Compute the following
  - (10 mod 2) / (1 mod 2)
  - (11 mod 3) / (2 mod 3)