# Number Theory & Cryptography

- Divisibility and Congruence
- Modular Arithmetic and its Applications
- GCD, (Extended) Euclidean Algorithm, Relative Prime
- The Caesar Cipher and Affine Cipher, Modular Inverse
- The Chinese Remainder Theorem
- Fermat's Little Theorem and Modular Exponentiation
- Private and Public Key Cryptography, The RSA Cryptosystem

# Cryptography

Cryptography is critical for secure communications on the Internet, privacy, integrity, and authentication

Cryptography encoding and decoding messages

- Cipher: A method for encoding messages
- Plaintext: The original message to be encoded
- Ciphertext: The encoded message
- Encryption: The process of encoding messages
- Decryption: The process of decoding messages

# Cryptography

Cryptography encoding and decoding messages

- In ancient Egypt, the first known use of encryption appeared in hieroglyphs, where simple ciphers were used to encode royal messages

- The evolution of cryptography is marked by key advancements, such as the Caesar cipher, the development of public-key systems, and the modern use of symmetric and asymmetric encryption methods

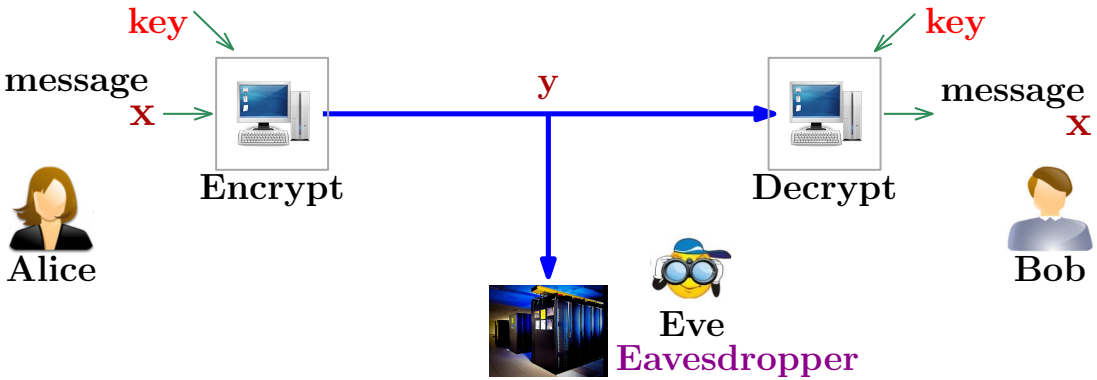
# Introduction to Security in Big Tech

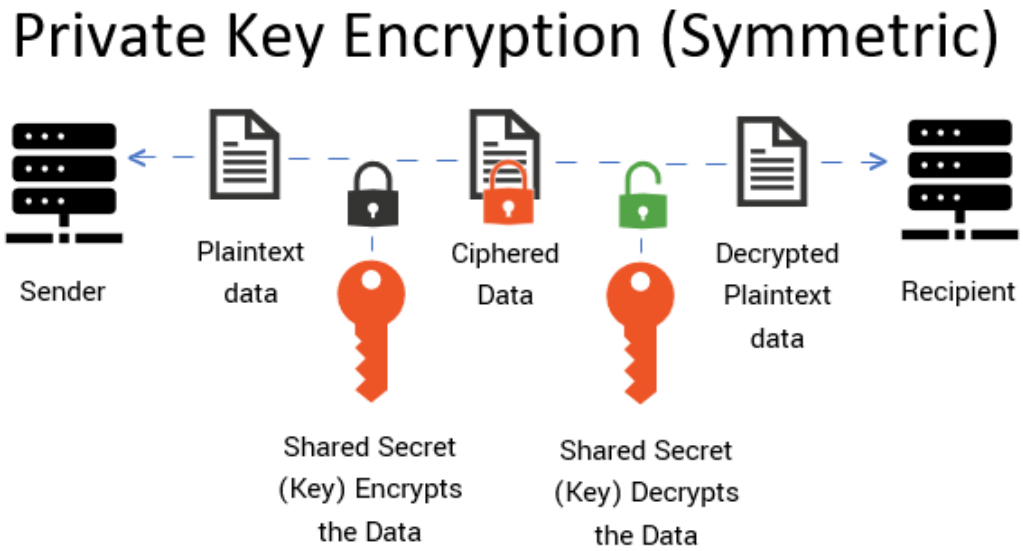
Cybersecurity has become a cornerstone of modern business operations in major tech companies. With massive user bases, sensitive data handling, and valuable digital assets, these companies are prime targets for cyberattacks

- Cyberattacks: Increasing frequency of breaches, phishing, and ransomware attacks
- Data Privacy: Critical to protect personal information, financial data, and intellectual property
- Legal Compliance: Regulations like GDPR, CCPA, and HIPAA impose strict data protection requirements
- Reputation Risk: Data breaches erode public trust and have major financial consequences

# Symmetric and Assymetric Encryption

Alice wants to send Bob a message, Eve is eavesdropping

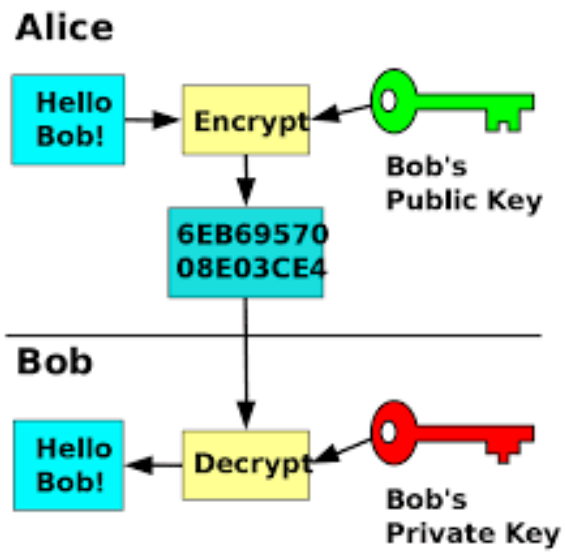


# Public Key Cryptography (PKI)

Public Key Infrastructure (PKI) allows secure communication over an insecure channel without pre-shared secret keys

- Public Key: Known to everyone; used to encrypt messages
- Private Key: Known only to the recipient; used to decrypt received messages
- Enables secure communication between strangers
- Foundation for secure web (HTTPS), cloud security, digital signatures, and secure emails



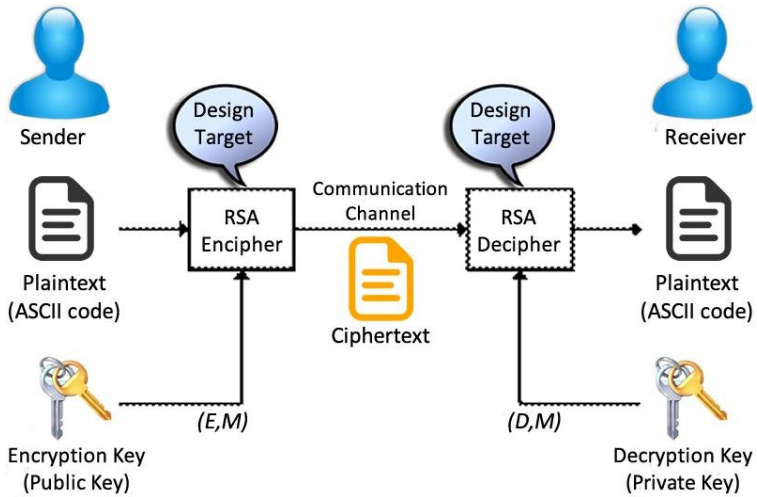

▷ Before PKI, securely exchanging keys was a major bottleneck!

# RSA Encryption

RSA (Rivest-Shamir-Adleman) was the first practical public-key cryptosystem, invented in 1977

- Based on the mathematical difficulty of factoring large prime numbers
- Forms the basis of secure web browsing (HTTPS), encrypted emails, digital signatures, and cloud security
- Still one of the most widely used encryption systems despite newer alternatives

▷ Fun fact: The original RSA algorithm was kept secret by MIT researchers until officially patented in 1983!



# Role of Encryption in Modern Tech

Encryption, including RSA, plays a pivotal role across Big Tech ecosystems:

- End-to-End Encryption: Ensures private conversations (e.g., WhatsApp, Gmail)
- Secure Transactions: Protects payment and financial data during online purchases
- Digital Signatures: Verifies sender identity and message integrity
- Cloud Security: Protects data stored and processed on cloud platforms

# Google's Use of RSA: Gmail Security

Google uses RSA encryption to secure its email platform, Gmail

- Secure Email Transmission:
  - RSA encrypts email during transmission over the internet (TLS)
  - Protects against interception by attackers
- Secure Attachments:
  - Attachments are encrypted, maintaining confidentiality
- Authentication and Identity Verification:
  - Public Key Certificates validate the identity of Gmail servers to users
  - Prevents users from connecting to malicious or fake servers during email transmission
- Secure Inter-Server Communication:
  - When Gmail exchanges emails with other providers (e.g., Yahoo, Outlook), RSA ensures encryption if the other party also supports it
  - Protects emails even when crossing into external networks

▷ Gmail's security warnings ("this message is not encrypted") appear when RSA/TLS is not properly used between servers

# Application of RSA in Amazon (E-Commerce Security)

RSA encryption underpins the security of Amazon's e-commerce ecosystem, especially during critical payment processes

- SSL/TLS Handshake: RSA is used in establishing secure HTTPS connections during checkout
  - Ensures that credit card information and personal data are encrypted during transmission
- Payment Gateway Security: Customer payment details are encrypted and securely transmitted to payment processors
  - Protects customers from man-in-the-middle attacks
- Compliance with Industry Standards: RSA encryption helps Amazon meet PCI-DSS standards for payment card security

# Application of RSA in WhatsApp (Meta)

RSA was used in WhatsApp end-to-end encryption for private communication

- Message Encryption: Messages are encrypted using the recipient's public key before transmission
  - Only the intended recipient can decrypt using their private key
- Call Encryption: Voice and video calls are protected through similar asymmetric encryption mechanisms
- Digital Signatures for Authentication: Each message carries a digital signature - verifies the sender's identity
  - Prevents impersonation or tampering of messages

▷ Although WhatsApp now mainly uses the Signal Protocol, RSA remains a core enabler in identity verification and initial key exchange

# RSA in Cloud Security (Google Cloud and AWS)

RSA encryption is a backbone of secure operations for major cloud service providers like Google Cloud and Amazon Web Services (AWS)

- Data-at-Rest Protection:
  - RSA encrypts sensitive files stored in cloud databases and storage buckets
- Secure API Communication:
  - API calls between cloud services are encrypted and authenticated using RSA certificates
- Identity and Access Management (IAM):
  - RSA public-private keys ensure that only authorized users can access cloud resources
  - Used in systems like AWS IAM Roles and Google Service Account

# Arithmetic Rules

- We will constrain the operands and results of basic arithmetic operations to a certain range — the modulus
- This is important for understanding cryptography, especially RSA

Assume arithmetic rules for operations $+, *, -$ on the set of integers

- $a(b+c) = ab+ac$
- $ab = ba$
- $a(bc) = (ab)c$
- $a*1 = a$
- $a*0 = 0$
- $a+0 = a$
- $a-a = 0$
- $a+1 > a$

# The divides operator

**Definition**

For $a, b \in \mathbb{Z}, a \neq 0$, we say $a \mid b$ : ($a$ divides $b$) if $\exists\, c \in \mathbb{Z} : b = ac$

$a$ divides $b$ if there is an integer $c$ such that $b = ac$

- $4 \mid 12$ $\triangleright$ $12 = 4 \cdot 3$
- $3 \mid 12$ $\triangleright$ $12 = 3 \cdot 4$
- $5 \mid 0$ $\triangleright$ $0 = 5 \cdot 0$
- $3 \nmid 7$

- $1 \mid 8$ $\triangleright$ $8 = 1 \cdot 8$
- $-2 \mid 6$ $\triangleright$ $6 = -2 \cdot -3$
- $-6 \mid -12$ $\triangleright$ $-12 = -6 \cdot 2$
- $-4 \nmid 13$

- $a$ is a factor or divisor of $b$
- $b$ is a multiple of $a$

# Divisibility Facts

Some useful properties of $\mid$ operator that can make calculations easier

1. $\forall n \quad 1 \mid n \quad \triangleright \quad n = 1 \cdot n$
2. $\forall n \quad n \mid n \quad \triangleright \quad n = n \cdot 1$
3. $\forall n \quad n \mid 0 \quad \triangleright \quad 0 = n \cdot 0$
4. $\forall n \quad -1 \mid n \quad \triangleright \quad n = -1 \cdot -n$
5. $\forall n \quad -n \mid n \quad \triangleright \quad n = -n \cdot -1$

# Divisibility Facts

## Theorem

*For* $a, b, c \in \mathbb{Z}$

1. $a \mid b \implies a \mid bc$
2. $a \mid b \wedge b \mid c \implies a \mid c$
3. $a \mid b \wedge a \mid c \implies a \mid b + c$

- $3 \mid 6 \implies 3 \mid 6 \cdot 2$
- $2 \mid 4 \wedge 4 \mid 8 \implies 2 \mid 8$
- $2 \mid 4 \wedge 2 \mid 8 \implies 2 \mid 8 + 4$

**Corollary:** $a \mid b \wedge a \mid c \implies a \mid mb + nc, \quad m, n \in \mathbb{Z}$

- $2 \mid 4 \wedge 2 \mid 8 \implies 2 \mid 3 \cdot 8 + 5 \cdot 4$

# Divisibility Facts

**Corollary:** $a \mid b \ \wedge \ a \mid c \implies a \mid mb + nc, \quad m, n \in \mathbb{Z}$

**Proof:** Number theory proofs generally use definition and basic arithmetic

$a \mid b \ \wedge \ a \mid c \implies \exists\, x, y : \quad b = ax \quad \wedge \quad c = ay$

$mb = m(ax) = a(mx) \implies a \mid mb$

$nc = n(ay) = a(ny) \implies a \mid nc$

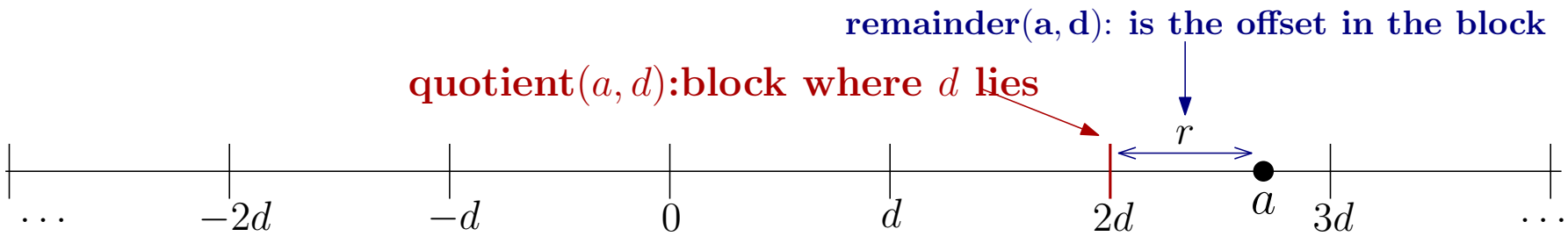
By Theorem part (2) $a \mid mb + nc$

# The Division Algorithm

## Theorem (The Division Algorithm)

*Let $a$ be an integer and $d$ a positive integer. Then there are unique integers $q$ and $r$, with $0 \leq r < d$ such that $a = dq + r$*

- $q$ : quotient$(a, d)$
- $r$ : remainder$(a, d)$ ▷ $a$ % $d$
- $d$ : divisor
- $a$ : dividend




Clearly with $a$ and $d > 0$, $q$ and $r$ are uniquely defined

## Congruence

For $a, b \in \mathbb{Z}$ and $m \in \mathbb{Z}^+$, $\quad a \equiv_m b \quad$ iff $\quad m \mid (a - b)$

pronounced as $\quad a$ is congruent to $b$ modulo $m$

▷ Standard notation for $a \equiv_m b$ is $a \equiv b \pmod{m}$

**Theorem:** Let $a, b \in \mathbb{Z}$ and $m \in \mathbb{Z}^+$.

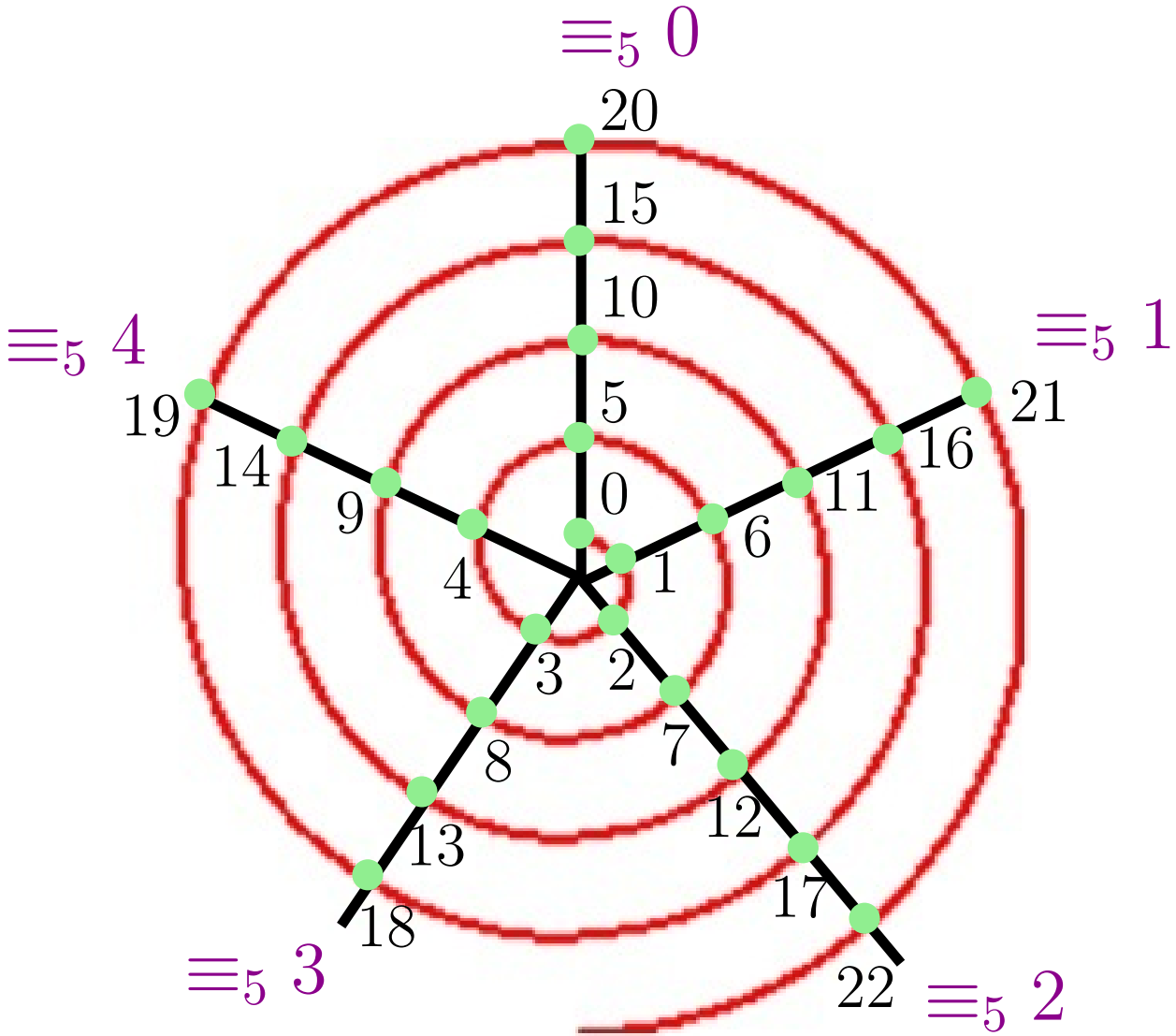
Then $a \equiv_m b \quad$ iff $\quad a \,\%\, m = b \,\%\, m$

$3 \equiv_3 6, \quad 3 \equiv_3 3, \quad 7 \equiv_5 2 \quad -3 \equiv_5 2, \quad -1 \equiv_3 -4$

To avoid confusion between standard notaitons - $\pmod{m}$ vs **mod** $m$, we use our notation.

Note that $\%\, m$ is an operator, while $\equiv_m$ is an equivalence relation over $\mathbb{Z}$

# Congruence


$\equiv_5 0$
20
15
10
5
0
$\equiv_5 4$
19
14
9
4
$\equiv_5 1$
21
16
11
6
1
3
2
8
7
13
12
18
17
$\equiv_5 3$
22
$\equiv_5 2$

# Congruence Facts

## Fact

1. $a \equiv_m a$
2. $a \equiv_m b \iff b \equiv_m a$
3. $a \equiv_m b \wedge b \equiv_m c \implies a \equiv_m c$

   $\triangleright$ *$\equiv_m$ is an equivalence relation on $\mathbb{Z}$*
4. $a \equiv_m (a \,\%\, m)$

**Theorem**

$a \equiv_m b \iff \exists\, k \in \mathbb{Z} : a = b + km$

- $8 \equiv_5 3$ and $8 = 3 + 5(1)$
- $16 \equiv_5 1$ and $16 = 1 + 5(3)$

**Proof:**

$a \equiv_m b$

$\leftrightarrow \; m|(a-b)$ ▷ by definition

$\leftrightarrow \; \exists\, k \in \mathbb{Z} : a - b = km$

$\leftrightarrow \; a = b + km$

# Congruence

**Definition**

For $a, b \in \mathbb{Z}$ and $m \in \mathbb{Z}^+$, $\quad a \equiv_m b \quad$ iff $\quad m \mid (a - b)$

**Theorem**

*For $a, b \in \mathbb{Z}$ and $m \in \mathbb{Z}^+$, $\quad a \equiv_m b \quad$ iff $\quad a \,\%\, m = b \,\%\, m$*

**Theorem**

*For $a, b \in \mathbb{Z}$ and $m \in \mathbb{Z}^+$, $\quad a \equiv_m b \iff \exists\, k \in \mathbb{Z} : a = b + km$*

# Modular Arithmetic

Modular Arithmetic rules are similar to the regular arithmetic rules but are applied to integers within a modular system

**Lemma**

*If* $a \equiv_m b$ *and* $c \equiv_m d$, *then* $a + c \equiv_m b + d$

▷ $8 \equiv_5 3$ and $9 \equiv_5 4 \implies 8 + 9 \equiv_5 3 + 4$

**Familiar cases:** $m = 2$ **and** $m = 10$

If $(a, b)$ and $(c, d)$ have the same parity, then $a + c$ and $b + d$ have the same parity

If $(a, b)$ and $(c, d)$ have the same last digit, then $a + c$ and $b + d$ have the same last digit

The lemma says it works for all $m$

**Lemma**

*If* $a \equiv_m b$ *and* $c \equiv_m d$, *then* $a + c \equiv_m b + d$

**Proof:** $a \equiv_m b \implies a = b + xm$ AND $c \equiv_m d \implies c = d + ym$

$$a + c = b + d + xm + ym \implies (a + c) - (b + d) = m(x + y)$$

Hence $m \mid (a + c) - (b + d)$

So $a + c \equiv_m b + d$

if $a \equiv b \pmod{m}$, then adding or multiplying both sides by the same number does not change the congruence

$(8 + 9) \mod 5$. First, we find that $8 \equiv 3 \pmod 5$ and $9 \equiv 4 \pmod 5$. Now, $8 + 9 \equiv 3 + 4 = 7 \equiv 2 \pmod 5$

**Lemma**

*If* $a \equiv_m b$ *and* $c \equiv_m d$, *then* $ac \equiv_m bd$

**Proof:**

Very similar!

**Lemma**

*If* $a \equiv_m b$, *then* $a^k \equiv_m b^k$

**Proof:**

Very similar!

# Modular Arithmetic

## Lemma

1. *If* $a \equiv_m b$ *and* $c \equiv_m d$, *then* $a + c \equiv_m b + d$
2. *If* $a \equiv_m b$ *and* $c \equiv_m d$, *then* $ac \equiv_m bd$
3. *If* $a \equiv_m b$, *then* $a^k \equiv_m b^k$

## Corollary

1. $(a + b) \,\%\, m = \big((a \,\%\, m) + (b \,\%\, m)\big) \,\%\, m$
2. $ab \,\%\, m = \big((a \,\%\, m)(b \,\%\, m)\big) \,\%\, m$
3. $a^k \,\%\, m = \big(a \,\%\, m\big)^k \,\%\, m$

This means that while computing $(a + c) \,\%\, m$ or $(ac) \,\%\, m$, we can replace $a$ with $(a \,\%\, m)$ and $c$ with $(c \,\%\, m)$ ▷ Recall that $a \equiv_m a \,\%\, m$

# Modular Arithmetic

## Lemma

1. *If* $a \equiv_m b$ *and* $c \equiv_m d$, *then* $a + c \equiv_m b + d$
2. *If* $a \equiv_m b$ *and* $c \equiv_m d$, *then* $ac \equiv_m bd$
3. *If* $a \equiv_m b$, *then* $a^k \equiv_m b^k$

## Corollary

1. $(a + b) \,\%\, m = \big((a \,\%\, m) + (b \,\%\, m)\big) \,\%\, m$
2. $ab \,\%\, m = \big((a \,\%\, m)(b \,\%\, m)\big) \,\%\, m$
3. $a^k \,\%\, m = \big(a \,\%\, m\big)^k \,\%\, m$

Compute $-706 \cdot 1456 \,\%\, 19$

$-706 \equiv_{19} 16$ and $1456 \equiv_{19} 12 \implies -706 \cdot 1456 \,\%\, 19 = 16 \cdot 12 \,\%\, 19$

# Modular Arithmetic

## Lemma

1. *If* $a \equiv_m b$ *and* $c \equiv_m d$, *then* $a + c \equiv_m b + d$
2. *If* $a \equiv_m b$ *and* $c \equiv_m d$, *then* $ac \equiv_m bd$
3. *If* $a \equiv_m b$, *then* $a^k \equiv_m b^k$

## Corollary

1. $(a + b) \,\%\, m = \big((a \,\%\, m) + (b \,\%\, m)\big) \,\%\, m$
2. $ab \,\%\, m = \big((a \,\%\, m)(b \,\%\, m)\big) \,\%\, m$
3. $a^k \,\%\, m = \big(a \,\%\, m\big)^k \,\%\, m$

$A = \{-706, 1456, 88, -41, 19, 20, 38, 40\}$ Compute $\left(\sum_{x \in A} x\right) \,\%\, 19$

Remainders: $R = \{16, 12, 12, 16, 0, 1, 0, 2\}$ So $\left(\sum_{x \in A} x\right) \,\%\, 19 = \left(\sum_{r \in R} r\right) \,\%\, 19$

# Modular Arithmetic

## Lemma

1. *If* $a \equiv_m b$ *and* $c \equiv_m d$, *then* $a + c \equiv_m b + d$
2. *If* $a \equiv_m b$ *and* $c \equiv_m d$, *then* $ac \equiv_m bd$
3. *If* $a \equiv_m b$, *then* $a^k \equiv_m b^k$

## Corollary

1. $(a + b) \,\%\, m = \big((a \,\%\, m) + (b \,\%\, m)\big) \,\%\, m$
2. $ab \,\%\, m = \big((a \,\%\, m)(b \,\%\, m)\big) \,\%\, m$
3. $a^k \,\%\, m = \big(a \,\%\, m\big)^k \,\%\, m$

Compute $516^{3031} \,\%\, 103$

$516 \equiv_{103} 1$ So $516^{3031} \,\%\, 103 = 1^{3031} \,\%\, 103 = 1$

**Theorem**

*A positive integer $N$ is divisible by 9 iff the sum of its digits is divisible by 9*

$9 \mid 343233153711$ because $9 \mid 3+4+3+2+3+3+1+5+3+7$

$9 \nmid 12356954236$ because $9 \nmid 1+2+3+5+6+9+5+4+2+3+6$

**Proof:** Note that $10 \equiv_9 1$

Let $N = d_k d_{k-1} \ldots d_2 d_1 d_0$ ▷ $d_i$ : $i^{th}$ digit of $N$

$$N = d_k 10^k + d_{k-1} 10^{k-1} + \cdots + d_2 10^2 + d_1 10^1 + d_0 10^0$$

Using the congruence identities

$$N \equiv_9 d_k 10^k + \ldots + d_2 10^2 + d_1 10^1 + d_0 10^0$$

$$N \equiv_9 d_k 1^k + \ldots + d_2 1^2 + d_1 1^1 + d_0 1^0$$

$$N \equiv_9 d_k + d_{k-1} + \ldots + d_2 + d_1 + d_0$$

# Modular Arithmetic: Applications

**Theorem**

*A positive integer $N$ is divisible by 3 iff the sum of its digits is divisible by 3*

**Proof:** Essentially the same

**Theorem**

*A positive integer $N$ is divisible by 11 iff the alternating sum of its digits is divisible by 11*

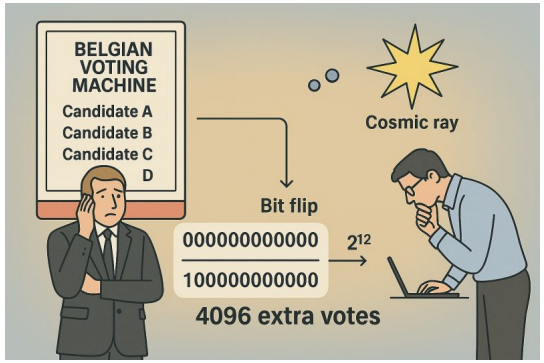
**Proof:** Essentially the same, using the fact that $10 \equiv_{11} -1$

# Modular Arithmetic: Applications

## Definition (Check Digit)

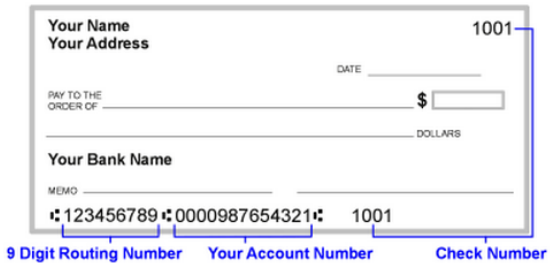
An extra digit appended to a number, which is related to the other digits in some way

▷ Catches most transposition and single-digit errors




**Airlines Tickets**


**Bank routing transit number**




12 digits ticket number, plus a $13^{th}$ check digit

check digit is the main number % 7

$01 - 1300696717 - 2$ as $11300696717 \% 7 = 2$

9-digits bank routing number

$d_8 d_7 \ldots d_3 d_2 d_1 d_0$ $d_0$ is check digit

$d_0 = 7d_8 + 3d_7 + 9d_6 + 7d_5 + 3d_4 + 9d_3 + 7d_2 + 3d_1 \% 10$

Difficult to find check digit by most calculators

Easier to compute using modular arithmetic

# Modular Exponentiation

Modular exponentiation is a key operation in RSA encryption. It allows us to calculate large powers of numbers under a modulus efficiently

Given (large) integers $b, m, n$ Find $b^n \% m$

Compute $2851^{3177} \% 4559$ ▷ $2851^{3177}$ has about $12k$ digits!

Find $22^4 \% 29$

Instead of calculating $b^n$ and then reducing modulo $m$, we can repeatedly reduce the intermediate results modulo $m$ after each multiplication

$$22^4 \% 29 = 22 \cdot 22 \cdot 22 \cdot 22 \% 29$$

$$= 22 \cdot 22 \cdot 484 \% 29 = 22 \cdot 22 \cdot 20 \% 29$$

$$= 22 \cdot 440 \% 29 = 22 \cdot 5 \% 29 = 110 \% 29 = 23$$

It helps for the number of digits (storage) but number of steps is still large – We will come back to it

# Prime Numbers

**Definition**

A positive integer $p$ is prime if it has exactly two divisors, namely 1 and $p$

1 is not prime

**Definition**

A positive integer $n$ is composite if it has a divisor $d$, $1 < d < n$

1 is not composite

- In cryptography, prime numbers play a crucial role, especially in RSA algorithm, where large primes are used to generate public and private keys
- In particular, the difficulty of factoring large composite numbers into their prime factors is the basis for the security of RSA encryption

# Greatest common divisor

$\text{GCD}(a, b) :=$ the greatest common divisor

▷ the largest integer $d$ that divides both $a$ and $b$

- $\text{GCD}(24, 32) = 8$
- $\text{GCD}(22, 24) = 2$
- $\text{GCD}(15, 5) = 5$
- $\text{GCD}(25, 15) = 5$
- $\text{GCD}(13, 20) = 1$
- $\text{GCD}(11, 33) = 11$

**Lemma:** $p$ is prime $\implies \forall\, a \in \mathbb{Z}$ $\text{GCD}(p, a) = 1$ or $p$

▷ $\because$ $p$ has only two divisors 1 and $p$

$a$ and $b$ are **relatively prime** if $\text{GCD}(a, b) = 1$

▷ Equivalently, $a$ and $b$ have no common factors

$\text{GCD}(25, 16) = 1, \quad \text{GCD}(24, 25) = 1$

A prime number $p$ is relatively prime to all integers except its multiples

# Computing GCD

$\text{GCD}(a, b) :=$ the greatest common divisor

$\triangleright$ the largest integer $d$ that divides both $a$ and $b$

Compute $\text{GCD}(a, b)$ by

1. find all divisors of $a$ and $b$
2. find the common divisors
3. find the greatest among the commons

Compute $\text{GCD}(a, b)$ from the prime factorization of $a$ and $b$

$$a = p_1^{a_1} p_2^{a_2} \dots p_n^{a_n} \qquad b = p_1^{b_1} p_2^{b_2} \dots p_n^{b_n}$$

$$\text{GCD}(a, b) = p_1^{\min\{a_1, b_1\}} p_2^{\min\{a_2, b_2\}} \dots p_n^{\min\{a_n, b_n\}}$$

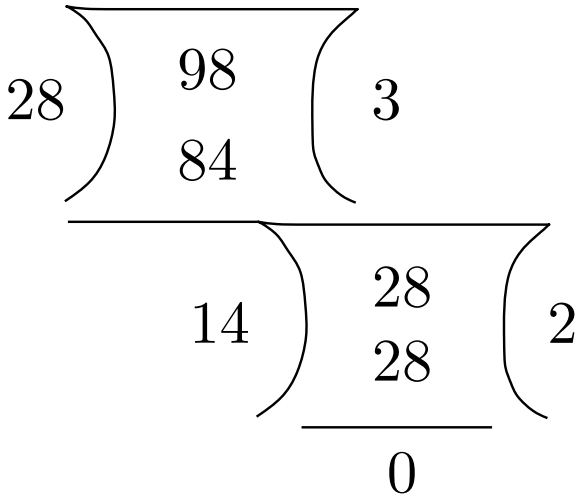
$$98 = 2 \cdot 7 \cdot 7 = 2^1\, 3^0\, 5^0\, 7^2\, 11^0 \cdots$$

$$420 = 2 \cdot 2 \cdot 3 \cdot 5 \cdot 7 = 2^2\, 3^1\, 5^1\, 7^1\, 11^0 \cdots$$

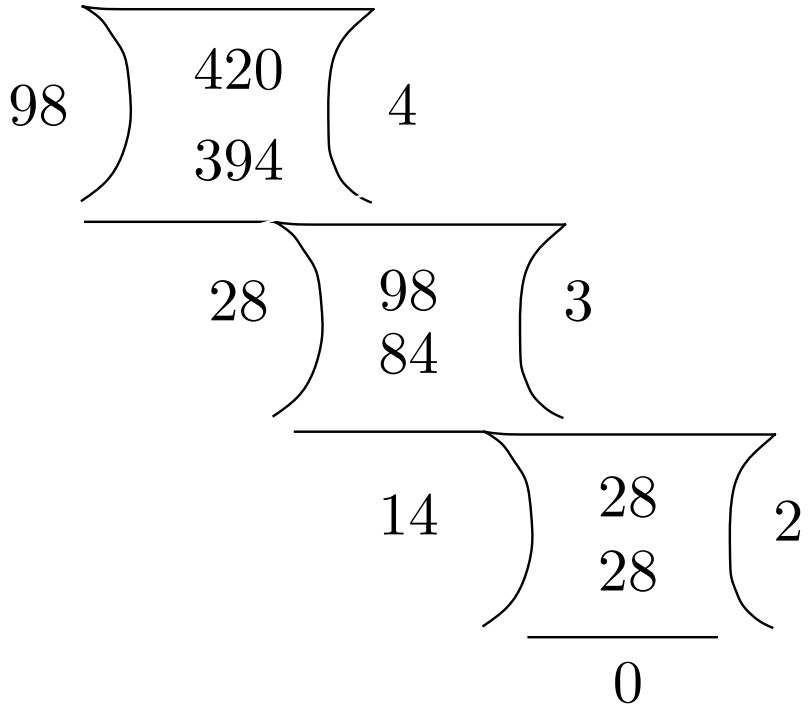
$$\text{GCD}(98, 420) = 2^1\, 3^0\, 5^0\, 7^1\, 11^0 \cdots = 14$$

# Computing GCD: Euclidean Algorithm

GCD$(28, 98)$

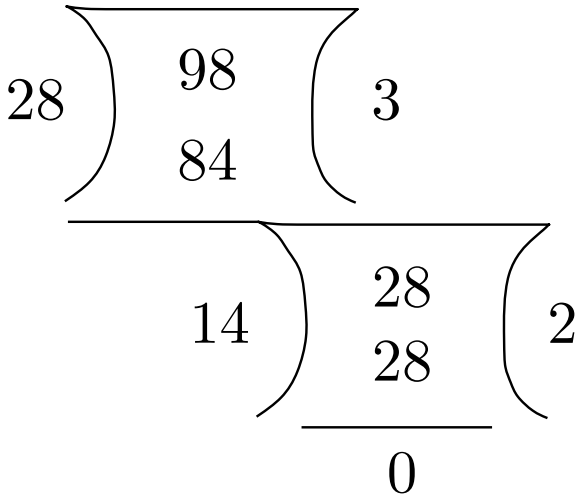



GCD$(98, 420)$

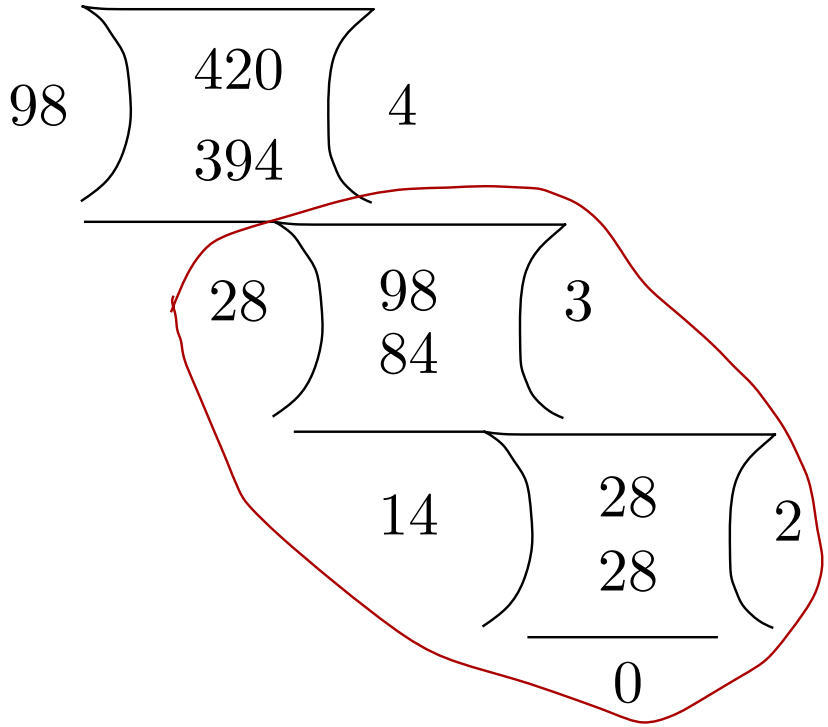


# Computing GCD: Euclidean Algorithm

$\text{GCD}(28, 98)$




$\text{GCD}(98, 420)$




**Theorem (Euclid)**

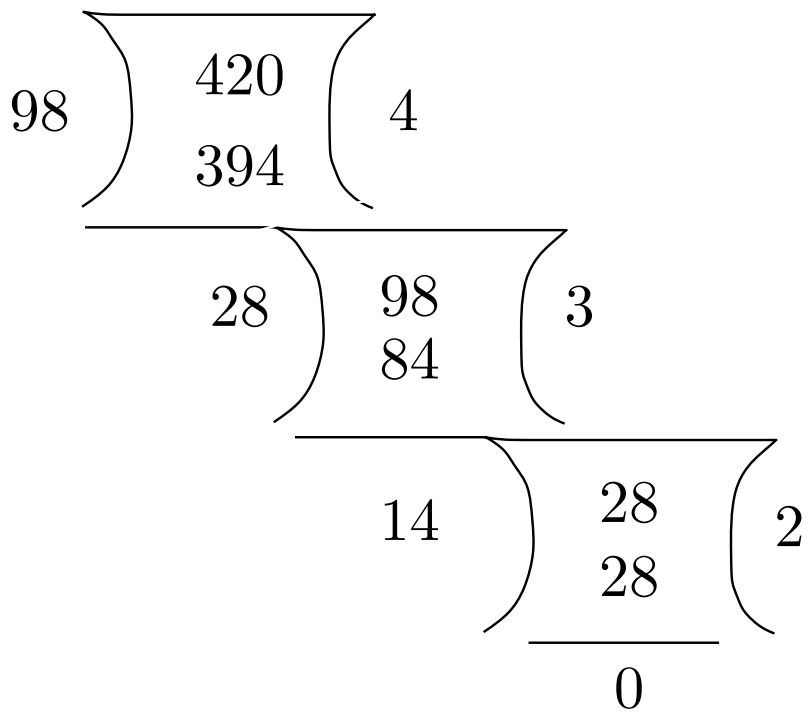
*If* $a = qb + r$, *then* $\text{GCD}(a, b) = \text{GCD}(b, r)$

# Computing GCD: Euclidean Algorithm

**Theorem (Euclid)**

*If* $a = qb + r$, *then* $\gcd(a, b) = \gcd(b, r)$

$\gcd(98, 420)$




$a = 420,\ b = 98$

$\triangleright\ 420 = 98 \cdot 4 + 28$

$\gcd(420, 98) = \gcd(98, 28)$

$\triangleright\ 98 = 28 \cdot 3 + 14$

$\gcd(98, 28) = \gcd(28, 14)$

$\triangleright\ 28 = 14 \cdot 2 + 0$

$\gcd(28, 14) = \gcd(14, 0) = 14$

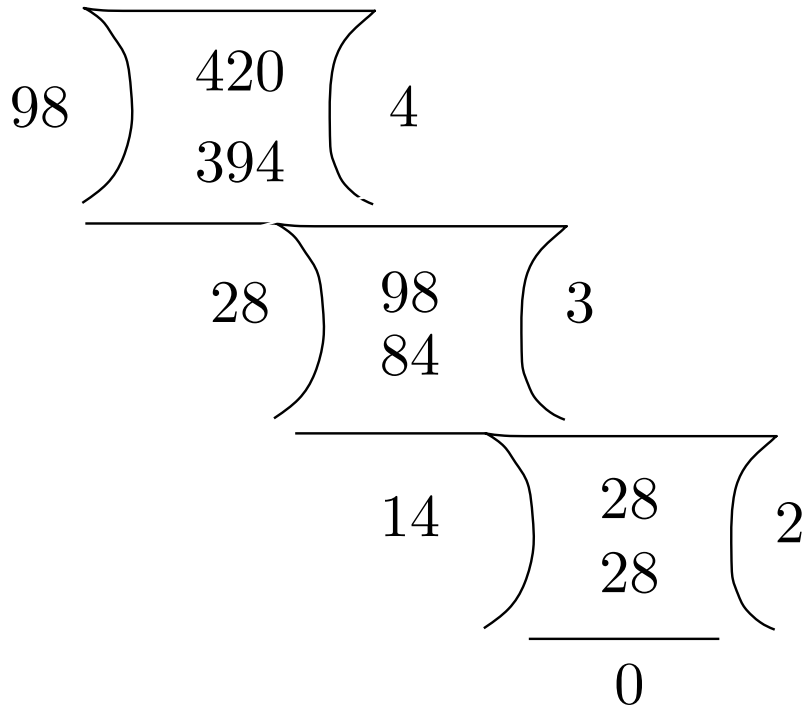
$\gcd(420, 98) = 14$

# Computing GCD: Euclidean Algorithm

**Theorem (Euclid)**

*If* $a = qb + r$, *then* $\gcd(a, b) = \gcd(b, r)$

$\gcd(98, 420)$




**Algorithm** GCD Computation

```
function GCD(a, b)
    if b = 0 then
        return a
    else
        r ← a % b
        return GCD(b, r)
```

# Computing GCD: Proof of the Euclidean Algorithm

**Theorem (Euclid)**

*If* $a = qb + r$, *then* $\gcd(a, b) = \gcd(b, r)$

**Proof:** **Case 1:** $r = 0 \implies \gcd(b, r) = \gcd(b, 0) = b$, as $b \mid 0$

$r = 0 \implies a = qb$, so $\gcd(a, b) = b = \gcd(b, r)$

**Case 2:** $r > 0$

Let $d$ be a common divisor of $b$ and $r$ $\quad b = xd$ and $r = yd$

$a = qb + r = (qx)d + yd = (qx + y)d \implies d \mid a$

Let $d$ be a common divisor of $a$ and $b$ $\quad a = sd$ and $b = td$

$r = a - qb = sd - (qt)d = (s + qt)d \implies d \mid r$

So $d$ is a common divisor of $a, b \leftrightarrow d$ is a common divisor of $b, r$

# GCD: Extended Euclidean Algorithm

**Theorem**

*For all $a, b$, $\exists\, s, t:\ sa + tb = \gcd(a, b)$*

$a = 420,\ b = 98$

$\triangleright\ 420 = 98 \cdot 4 + 28$

$\gcd(420, 98) = \gcd(98, 28)$

$\triangleright\ 98 = 28 \cdot 3 + 14$

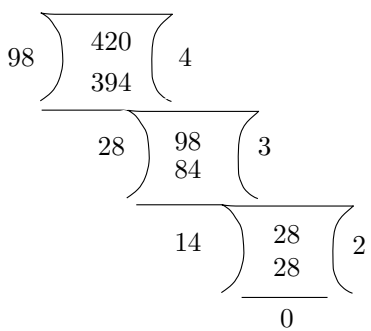
$\gcd(98, 28) = \gcd(28, 14)$

$\triangleright\ 28 = 14 \cdot 2 + 0$

$\gcd(28, 14) = \gcd(14, 0) = 14$

$\gcd(420, 98) = 14$

$\gcd(98, 420)$




$\gcd(420, 98) = 14$

$\triangleright\ 14 = 98 - 3 \cdot 28$

$\gcd(420, 98) = 98 - 3 \cdot 28$

$\triangleright\ 28 = 420 - 98 \cdot 4$

$\gcd(420, 98) = 98 - 3(420 - 4 \cdot 98)$

$\gcd(420, 98) = -3 \cdot 420 + 13 \cdot 98$

$s = -3, \quad t = 13$

# GCD: Extended Euclidean Algorithm

**Theorem**

*For all $a, b$, $\exists\, s, t:\ sa + tb = \gcd(a, b)$*

$a = 899,\ b = 493$

$\triangleright\ 899 = 1 \cdot 493 + 406$

$\gcd(899, 493) = \gcd(493, 406)$

$\triangleright\ 493 = 1 \cdot 406 + 87$

$\gcd(493, 406) = \gcd(406, 87)$

$\triangleright\ 406 = 4 \cdot 87 + 58$

$\gcd(406, 87) = \gcd(87, 58)$

$\triangleright\ 87 = 1 \cdot 58 + 29$

$\gcd(87, 58) = \gcd(58, 29)$

$\triangleright\ 58 = 2 \cdot 29 + 0$

$\gcd(58, 29) = \gcd(29, 0) = 29$

$\gcd(899, 493) = 29$

$29 = 87 - 1 \cdot 58$

$\triangleright\ 58 = 406 - 4 \cdot 87$

$29 = 87 - 1(406 - 4 \cdot 87)$

$\triangleright\ 87 = 493 - 1 \cdot 406$

$29 = 5(493 - 406) - 406$

$\triangleright\ 406 = 899 - 1 \cdot 493$

$29 = 5 \cdot 493 - 6(899 - 493)$

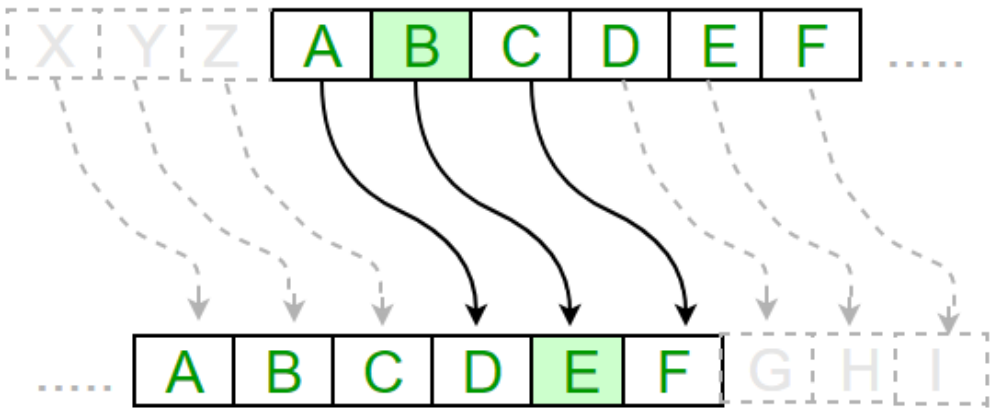
$29 = -6 \cdot 899 + 11 \cdot 493$

$s = -6,\quad t = 11$

# The Caesar Cipher

A substitution cipher: replaces each letter in the plaintext with another letter according to a fixed system

The Caesar Cipher (a special case of substitution): Substitute each letter by the letter a fixed number of places (say 3) down the alphabet




How about $x, y$ and $z$?

▷ Cyclic–modular

# The Caesar Cipher

A substitution cipher: replaces each letter in the plaintext with another letter according to a fixed system

The Caesar Cipher (a special case of substitution): Substitute each letter the letter a fixed number of places (say 3) down the alphabet

Replace 3 with some other integer $s$

**Encryption**

$c \leftarrow (p + s) \% 26$

**Decryption**

$p \leftarrow (c - s) \% 26$

For a Caesar cipher with a shift of 3 ($s = 3$):

Plaintext: HELLO  Ciphertext: KHOOR

Here, H becomes K, E becomes H, L becomes O, and O becomes R

# Affine Cipher

Affine Cipher: An extension of the Caesar cipher. Instead of shifting letters by a fixed amount, we apply an affine transformation

**Encryption**

$$c \leftarrow (tp + s) \,\%\, 26$$

**Decryption**

$$p \leftarrow \frac{(c - s)}{t} \,\%\, 26$$

With $t = 5$, $s = 8$, and plaintext "HELLO":

$$\mathrm{H} \to 7, \quad \mathrm{E} \to 4, \quad \mathrm{L} \to 11, \quad \mathrm{L} \to 11, \quad \mathrm{O} \to 14$$

Then applying the affine cipher formula, we get the ciphertext

$$tp = (c - s) \,\%\, 26 \quad \Longrightarrow \quad p = t^{-1}(c - s) \,\%\, 26$$

If $t = 3$, then $3 \cdot 9 = 27 \,\%\, 26 = 1$ ▷ $9 = 3^{-1}$

If $t = 5$, then $5 \cdot 21 = 105 \,\%\, 26 = 1$ ▷ $21 = 5^{-1}$

Not every integer has an inverse What is inverse of 4 modulo 26?

# Modular Inverse

The modular inverse is an important concept for decryption in RSA

**Definition**

$b$ is the inverse of $a$ modulo $m$ iff $a \times b \equiv_m 1$

For real numbers, every $x \neq 0 \in \mathbb{R}$ has an inverse

For integers, only 1 has an inverse

What if we were doing modular arithmetic?

Interesting property: integers also have inverses (at least some of them)

Find the modular inverse of 3 modulo 7

Need to find a number $b$ such that $3 \times b \equiv 1 \pmod 7$

We find that $3 \times 5 = 15 \equiv 1 \pmod 7$, so the inverse of 3 modulo 7 is 5

# Modular Inverse

## Definition

$b$ is the inverse of $a$ modulo $m$ iff $a \cdot b \equiv_m 1$

| $Z_5$ | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 1 | 1 | 2 | 3 | 4 |
| 2 | 2 | 4 | 1 | 3 |
| 3 | 3 | 1 | 4 | 2 |
| 4 | 4 | 3 | 2 | 1 |

| $Z_6$ | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| 1 | 1 | 2 | 3 | 4 | 5 |
| 2 | 2 | 4 | 0 | 2 | 4 |
| 3 | 3 | 0 | 3 | 0 | 3 |
| 4 | 4 | 2 | 0 | 4 | 2 |
| 5 | 5 | 4 | 3 | 2 | 1 |

# Modular Inverse

**Definition**

*b* is the inverse of *a* modulo *m* iff $a \times b \equiv_m 1$

**Theorem**

*a has an inverse modulo m iff a and m are relatively primes*

Equivalently, inverse of *a* modulo *m* exists iff $\text{GCD}(a, m) = 1$

$\text{GCD}(3, 7) = 1$ $\quad$ $3 \cdot 5 \,\%\, 7 = 1$

$\text{GCD}(4, 11) = 1$ $\quad$ $4 \cdot 3 \,\%\, 11 = 1$

$\text{GCD}(8, 9) = 1$ $\quad$ $8 \cdot 8 \,\%\, 9 = 1$

# Modular Inverse

**Theorem**

*a has an inverse modulo m iff* $\mathrm{GCD}(a, m) = 1$

**Proof:**

$\mathrm{GCD}(a, m) = 1$

$\implies sa + tm = 1$

$\implies tm = 1 - sa \implies m \mid 1 - sa$

$\implies 1 - sa \equiv_m 0$

$\implies sa \equiv_m 1$

We can find $s$ and $t$ from Extended Euclidean Algorithm

# Modular Arithmetic: Cancellation

If $a \equiv_m b$, then $a + c \equiv_m b + c$

If $a \equiv_m b$, then $ac \equiv_m bc$

Just as in '$=$' for real numbers

if $ac \equiv_m bc$, then IS $a \equiv_m b$?

$3 \cdot 4 \equiv_8 1 \cdot 4$ but $3 \not\equiv_8 1$

$4 \cdot 3 \equiv_9 1 \cdot 3$ but $4 \not\equiv_9 1$

$2 \cdot 4 \equiv_{12} 5 \cdot 4$ but $2 \not\equiv_{12} 5$

**We cannot cancel two "equal" values on both side of a congruence**

# Modular Arithmetic: Cancellation

**Lemma**

*Let* $\text{GCD}(a, m) = 1$. *If* $ab \equiv_m ac$, *then* $b \equiv_m c$

$$\text{GCD}(a, m) = 1 \implies \exists\, a^{-1} : aa^{-1} \equiv_m 1$$

$$ab \equiv_m ac$$

$$\implies aba^{-1} \equiv_m aca^{-1}$$

$$\implies b \equiv_m c$$

Typically modulus is a prime $\implies$ an inverse exists for every integer

**Modulo a prime, integers behave "like" real numbers**

## Solving Congruence

Finding $a^{-1}$ % $m$ is solving the congruence $ax \equiv_m 1$

How about solving other congruences!

$$\text{Solve } 2x \equiv_7 3$$

$\text{GCD}(2,7) = 1$ and $2 \cdot 4 \equiv_7 1$ so $4$ is $2^{-1}$

$2x \equiv_7 3 \implies 2x \cdot 4 \equiv_7 3 \cdot 4$

$\implies x \equiv_7 12 \equiv_7 5$

Verify that all integers of the form $5 + 7t$ satisfy this congruence

## Solving Congruence

Finding $a^{-1} \% m$ is solving the congruence $ax \equiv_m 1$

How about solving other congruences!

$$\text{Solve } 3x \equiv_6 2$$

Going through all numbers $\% 6$, no $x$ satisfy this congruence

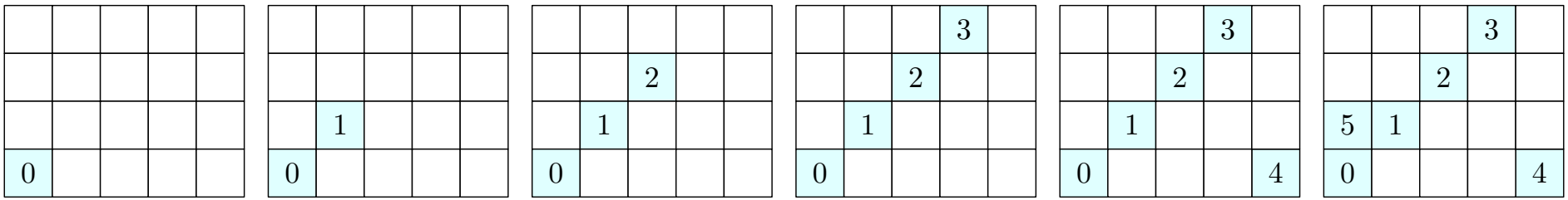
We say

$$3x \equiv_6 2 \quad \text{has no solutions}$$

# The Chinese Remainder Theorem

## The Chinese remainder theorem characterizes solvable system of simultaneous congruences and derive a solution

- Make an $m \times n$ grid
- Start from lower left and move up and right
- Wrap around both from top to bottom and right to left
- At every step write integers starting from 0




| | | | | |
|---|---|---|---|---|
| 15 | 11 | 7 | 3 | 19 |
| 10 | 6 | 2 | 18 | 14 |
| 5 | 1 | 17 | 13 | 9 |
| 0 | 16 | 12 | 8 | 4 |

# The Chinese Remainder Theorem

- Make an $m \times n$ grid
- Start from lower left and move up and right
- Wrap around both from top to bottom and right to left
- At every step write integers starting from 0

| | | | | | |
|---|---|---|---|---|---|
| | 7 | | 3 | | 11 |
| 6 | | 2 | | 10 | |
| | 1 | | 9 | | 5 |
| 0 | | 8 | | 4 | |

# The Chinese Remainder Theorem

- Make an $m \times n$ grid
- Start from lower left and move up and right
- Wrap around both from top to bottom and right to left
- At every step write integers starting from 0
- For which $m$ and $n$ the grid gets completely filled in?

| | | | | |
|---|---|---|---|---|
| 15 | 11 | 7 | 3 | 19 |
| 10 | 6 | 2 | 18 | 14 |
| 5 | 1 | 17 | 13 | 9 |
| 0 | 16 | 12 | 8 | 4 |

| | | | | | |
|---|---|---|---|---|---|
| | 7 | | 3 | | 11 |
| 6 | | 2 | | 10 | |
| | 1 | | 9 | | 5 |
| 0 | | 8 | | 4 | |

# The Chinese Remainder Theorem

Anceint Tale: In a war some soldiers died, wanted to find how many soldiers (**x**) are left. The Chinese emperor ordered a series of tasks

# The Chinese Remainder Theorem

Anceint Tale: In a war some soldiers died, wanted to find how many soldiers (**x**) are left. The Chinese emperor ordered a series of tasks

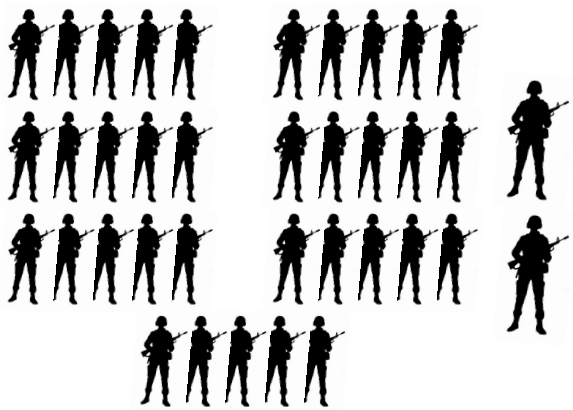
**Task-1:** Make groups of 3 and report how many couldn't $\triangleright\ x\ \%\ 3 = 1$

# The Chinese Remainder Theorem

Anceint Tale: In a war some soldiers died, wanted to find how many soldiers (**x**) are left. The Chinese emperor ordered a series of tasks

**Task-1:** Make groups of 3 and report how many couldn't $\triangleright\ x \,\%\, 3 = 1$

**Task-2:** Make groups of 5 and report how many couldn't $\triangleright\ x \,\%\, 5 = 2$



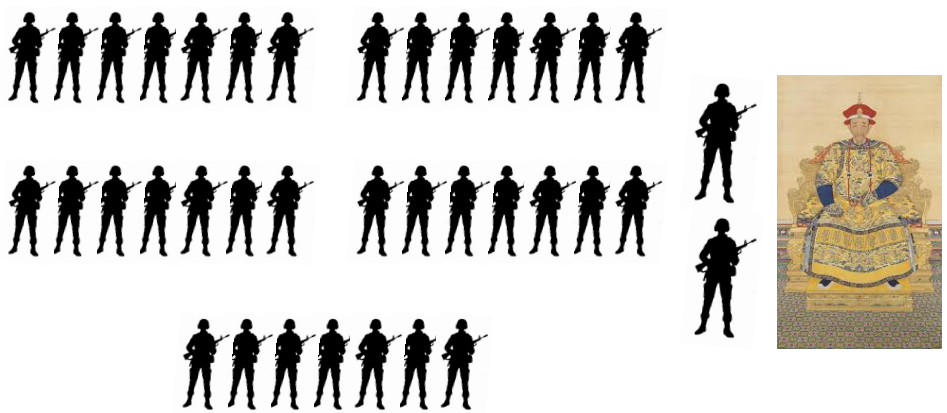
# The Chinese Remainder Theorem

Anceint Tale: In a war some soldiers died, wanted to find how many soldiers (**x**) are left. The Chinese emperor ordered a series of tasks

**Task-1:** Make groups of 3 and report how many couldn't $\triangleright\ x \,\%\, 3 = 1$

**Task-2:** Make groups of 5 and report how many couldn't $\triangleright\ x \,\%\, 5 = 2$

**Task-3:** Make groups of 7 and report how many couldn't $\triangleright\ x \,\%\, 7 = 2$



# The Chinese Remainder Theorem

Anceint Tale: In a war some soldiers died, wanted to find how many soldiers (**x**) are left. The Chinese emperor ordered a series of tasks

**Task-1:** Make groups of 3 and report how many couldn't ▷ $x \,\%\, 3 = 1$

**Task-2:** Make groups of 5 and report how many couldn't ▷ $x \,\%\, 5 = 2$

**Task-3:** Make groups of 7 and report how many couldn't ▷ $x \,\%\, 7 = 2$

Magically the emperor figured out their number ▷ $\mathbf{x = 37}$

# The Chinese Remainder Theorem

Anceint Tale: In a war some soldiers died, wanted to find how many soldiers (**x**) are left. The Chinese emperor ordered a series of tasks

Magically the emperor figured out their number ▷ $\mathbf{x = 37}$

Solve a system of modular congruences.

Find $x \leq 3 \cdot 5 \cdot 7$ satisfying

$$\begin{aligned} x &\equiv_3 1 \\ x &\equiv_5 2 \\ x &\equiv_7 2 \end{aligned}$$

**Theorem**

*If $m_1, m_2, m_3$ are relatively prime and $a_1, a_2, a_3$ are integers, then*

$$\begin{aligned} x &\equiv_{m_1} a_1 \\ x &\equiv_{m_2} a_2 \\ x &\equiv_{m_3} a_3 \end{aligned}$$

*has a unique solution modulo $m = m_1 m_2 m_3$*

**Proof by construction:**

1. $n_1 = m/m_1$
2. $y_1 = n_1^{-1} \% m_1$

1. $n_2 = m/m_2$
2. $y_2 = n_2^{-1} \% m_2$

1. $n_3 = m/m_3$
2. $y_3 = n_3^{-1} \% m_3$

▷ $y_k$ always exists as $\mathrm{GCD}(n_k, m_k) = 1$

$x = a_1 n_1 y_1 + a_2 n_2 y_2 + a_3 n_3 y_3$

$x$ satisfies all congruences. Uniqueness!

# The Chinese Remainder Theorem

Solve the system of modular congruences

$$x \equiv_3 1$$
$$x \equiv_5 2$$
$$x \equiv_7 2$$

Find $n_1, y_1, n_2, y_2, n_3, y_3$ as follows

$$n_1 = 5 \times 7 = 35 \quad y_1 = 35^{-1} \text{ modulo } 3 = 2$$
$$n_2 = 3 \times 7 = 21 \quad y_2 = 21^{-1} \text{ modulo } 5 = 1$$
$$n_3 = 3 \times 5 = 15 \quad y_3 = 15^{-1} \text{ modulo } 7 = 1$$

Note that by construction

$$n_1 y_1 \equiv_3 1, \quad n_1 y_1 \equiv_5 0, \quad n_1 y_1 \equiv_7 0$$
$$n_2 y_2 \equiv_3 0, \quad n_2 y_2 \equiv_5 1, \quad n_2 y_2 \equiv_7 0$$
$$n_3 y_3 \equiv_3 0, \quad n_3 y_3 \equiv_5 0, \quad n_3 y_3 \equiv_7 1$$

$$x = a_1 n_1 y_1 + a_2 n_2 y_2 + a_3 n_3 y_3 = 1 \cdot 70 + 2 \cdot 21 + 2 \cdot 15 = 142 \equiv_{105} 37$$

Verify that $37 \equiv_3 1, \quad 37 \equiv_5 2, \quad 37 \equiv_7 2$

# The Chinese Remainder Theorem

## Theorem

*If $m_1, m_2, \ldots, m_n$ are relatively prime and $a_1, a_2, \ldots, a_n$ are integers, then*

$$\begin{aligned} x &\equiv_{m_1} a_1 \\ x &\equiv_{m_2} a_2 \\ &\vdots \\ x &\equiv_{m_n} a_n \end{aligned}$$

*has a unique solution modulo $m = \prod_{i=1}^{n} m_i$*

Proof by construction is the same

# The Chinese Remainder Theorem

Using CRT we can uniquely represent any integer with remainders when moduli are relatively prime

▷ The integer has to be less than the product of moduli

Any integer $0 \leq x < 15$ can be represented by $(x \,\%\, 3, x \,\%\, 5)$

$12 = (0, 2)$
$11 = (2, 1)$

How many ordered pairs are possible? ▷ Will the grid fill?

Used two smaller integers to represent a big integer!

To perform arithmetic upon large integers, we can instead perform arithmetic on these small remainders

# The Chinese Remainder Theorem

Compute $123684 + 413456$

By CRT any $0 \leq x < 99 \cdot 98 \cdot 97 \cdot 95 = 89,403,930$ can be represented by its remainders modulo these moduli

$$123684 + 413456 = (33, 8, 9, 89) + (32, 92, 42, 16)$$

$$123684 + 413456 = (65, 2, 51, 10)$$

To convert back, Solve

$$x \equiv_{99} 65$$
$$x \equiv_{98} 2$$
$$x \equiv_{97} 51$$
$$x \equiv_{95} 10$$

We get

$$x = 123684 + 413456 = 537140$$

# The Chinese Remainder Theorem

Compute $1345 \times 2368$

By CRT any $0 \leq x < 99 \cdot 98 \cdot 97 \cdot 95 = 89,403,930$ can be represented by its remainders modulo these moduli

$1345 \times 2368$

$= (58, 71, 84, 15) * (91, 16, 40, 88)$ ▷ coordinate-wise products

$= (5278, 1136, 3360, 1320) \ = \ (31, 58, 62, 85)$ ▷ Took mod

To convert back, Solve

$$x \equiv_{99} 31$$
$$x \equiv_{98} 58$$
$$x \equiv_{97} 62$$
$$x \equiv_{95} 85$$

We get

$$x \ = \ 1345 \times 2368 \ = \ 3184960$$

# Pseudoprimes

**Theorem (Ancient Chinese)**

*Let $n$ be a prime, then $2^{n-1} \equiv_n 1$*

Some thought that the converse was also true!

The converse is not true!

$2^{340} \equiv_{341} 1$, but $341 = 31 \cdot 11$

Composites having this property are called pseudoprimes

# Fermat's Little Theorem

**Theorem (Ancient Chinese)**

*Let $n$ be a prime, then $2^{n-1} \equiv_n 1$*

**Theorem (Fermat's Little Theorem)**

*Let $p$ be a prime, then*

- *If $p \nmid a$, then $a^{p-1} \equiv_p 1$*
- *$a^p \equiv_p a$, for every integer $a$*

**Proof:** WLOG, assume that $a \in \{0, 1, \ldots, p-1\}$

If $a = 0$, this trivially holds

If $a \neq 0$, then $\gcd(a, p) = 1 \implies a$ has a multiplicative inverse, modulo $p$

Multiplying both sides by this inverse $a^{-1}$ yields: $a^{p-1} \equiv 1 \pmod{p}$

Fermat's Little Theorem is very useful for simplifying certain large exponentiations

# Fermat's Little Theorem

**Fermat's Little Theorem:** Let $p$ be a prime, If $p \nmid a$, then $a^{p-1} \equiv_p 1$

$p = 11$

| $a$ | $a^1$ | $a^2$ | $a^3$ | $a^4$ | $a^5$ | $a^6$ | $a^7$ | $a^8$ | $a^9$ | $a^{10}$ |
|---|---|---|---|---|---|---|---|---|---|---|
| 2 | 2 | 4 | 8 | 5 | 10 | 9 | 7 | 3 | 6 | 1 |
| 3 | 3 | 9 | 5 | 4 | 1 | 3 | 9 | 5 | 4 | 1 |
| 4 | 4 | 5 | 9 | 3 | 1 | 4 | 5 | 9 | 3 | 1 |
| 5 | 5 | 3 | 4 | 9 | 1 | 5 | 3 | 4 | 9 | 1 |
| 6 | 6 | 3 | 7 | 9 | 10 | 5 | 8 | 4 | 2 | 1 |
| 7 | 7 | 5 | 2 | 3 | 10 | 4 | 6 | 9 | 8 | 1 |
| 8 | 8 | 9 | 6 | 4 | 10 | 3 | 2 | 5 | 7 | 1 |
| 9 | 9 | 4 | 3 | 5 | 1 | 9 | 4 | 3 | 5 | 1 |
| 10 | 10 | 1 | 10 | 1 | 10 | 1 | 10 | 1 | 10 | 1 |

# Fermat's Little Theorem

**Fermat's Little Theorem:** Let $p$ be a prime, If $p \nmid a$, then $a^{p-1} \equiv_p 1$

- $p = 11$, $a^{10} = 1$ for all $a$ ▷ [FLT]
- For some $a$'s the exponent gets to 1 before $10 = p - 1$
- Patterns are of lengths that divides 10
- The values $a$ for which all numbers $1 \leq k \leq 10$ appear are called **generators**

# Fermat's Little Theorem: Modular Exponentiation

Given int $b$ and ints $n, p \geq 1$, find $b^n \% p$

When modulus is prime, we use FLT to speed up

Find $22^{61} \% 29$

$$22^{61} = 22^{28+28+5} = 22^{28} \cdot 22^{28} \cdot 22^5 = 1 \cdot 1 \cdot 22^5$$

$b^k \% p$ repeats after $k$ reaches $p-1$, so we use

$$b^n \% p = b^{n\%(p-1)} \% p$$

$$22^{61} \% 29 = 22^{61\%28} \% 29 = 22^5 \% 29$$

# Fermat's Little Theorem: Modular Exponentiation

Fermat's Little Theorem is very useful for simplifying certain large exponentiations

$7^{1027} =$

$$
\begin{aligned}
&823544544599327001495546228471606925197566190230622324273960155843749995838247324299808795643737413143459304292037082481398609160820269503301672056029937808578799506374779881698816017119148232704767843317\\
&101102037987772914665213147759018383011564881827311366784706944723048938602116196081664568260010752326060139580331874451190409068034895183321084488463006318582294519193149813795294091072551244801135441743\\
&892785357786574716992541099898150642976554895446350835310499206218444525020034477269414034632300034083364138440845589764562620181349309\\
&708422226147878465831533275981986254247462071986815725524826565630320246326497626307100615402346632648198420747422592562191688628689566677101890054018914525531500897548585341110302017650695463976958126547\\
&339816655368896053289891660447938682318914714384746873109524774725567321851877420158736581518028903696311777623045939543 \% 13 = 6
\end{aligned}
$$

$$7^{1027} \% 13 = 7^{1027\%12} \% 13 = 7^7 \% 13 = 823543 \% 13 = 6$$

## Generating Large Prime Numbers

To implement RSA, we need large prime numbers. A "guess and check" heuristic works due to of the following number-theoretic results

**Theorem (Prime Number Theorem)**

*The probability that a random number $n$ is prime is $\sim 1/(\ln n)$, i.e.,*

$$\lim_{n\to\infty} (\textit{proportion of numbers } \leq n \textit{ that are prime}) - \frac{1}{\ln n} = 0$$

- The chances of a random 9-digit number being prime is $\sim 4\%$ (i.e., 1 in 25). For a 200-digit number, this is approx. 0.2% (i.e., 1 in 500)

**Algorithm 2** Generate Large Prime Number

```
while true do
    n ← a random 200-digit number
    if n is prime then                    ▷ Need primality testing
        return n
```

## Checking Whether a Large Number is Prime

The Fermat primality test is a probabilistic method to determine whether a number is ("probably") prime

Fermat's Little Theorem: Let $p$ be a prime, If $p \nmid a$, then $a^p \equiv_p a$

**Algorithm 3** Fermat Primality Test $n \in \mathbb{N}$

1: Compute $a^{n-1} \pmod{n}$ for *many* random values of $a < n$
2: **if** for some $a$, $a^{n-1} \not\equiv 1 \pmod{n}$ **then**
3: $n$ must be composite
4: **else**
5: $n$ is "probably prime"

# Fermat Primality Test

## Fermat Primality Test (Revisited)

Given a number $n \in \mathbb{N}$, compute $a^{n-1} \pmod{n}$ for many random values of $a < n$

- If $a^{n-1} \not\equiv 1 \pmod{n}$, then $n$ must be composite. $\triangleright$ $a$ is a Fermat witness
- If $a^{n-1} \equiv 1 \pmod{n}$, there are two cases:
  1. $n$ is prime
  2. $n$ is composite $\triangleright$ $a$ is called a Fermat liar

## Lemma

If a composite number $n$ has a Fermat witness, then at least half of all numbers $a \in \{1, 2, \ldots, n-1\}$ that are relatively prime to $n$ are Fermat witnesses to $n$

Proof (Sketch): Let $a$ and $b$ be a Fermat witness and a Fermat liar for $n$

$$(ab)^{n-1} = \underbrace{a^{n-1}}_{\not\equiv 1} \cdot \underbrace{b^{n-1}}_{\equiv 1} \equiv a^{n-1} \not\equiv 1 \pmod{n}$$

In other words, every Fermat liar $b$ has a corresponding Fermat witness $ab$

## Carmichael Numbers

We saw that if $n$ has a Fermat witness, then it has many

Is it possible that $n$ is composite, but has no Fermat witnesses?

Unfortunately, the answer is YES, but this is very rare

A **Carmichael number** is a composite number $n$ for which

$$a^{n-1} \equiv 1 \pmod{n}$$

holds for all $a = 1, \ldots, n-1$ relatively prime to $n$

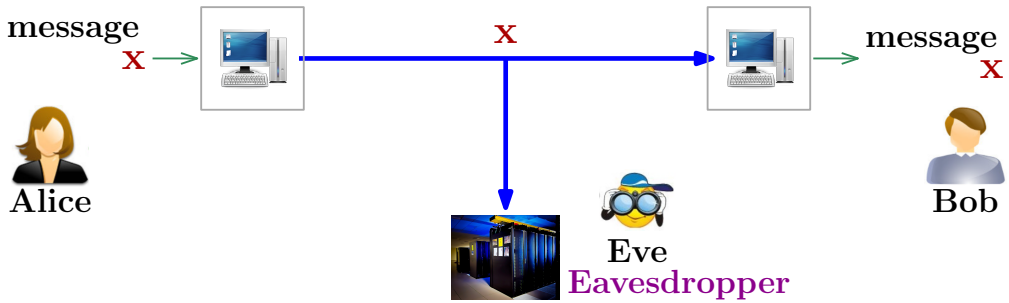
First few Carmichael numbers: 561, 1105, 1729, 2465, 2821, 6601, 8911

For 100-digit numbers, less than 1 in $10^{30}$ are Carmichael numbers. For 200-digit numbers, the chances are even less.

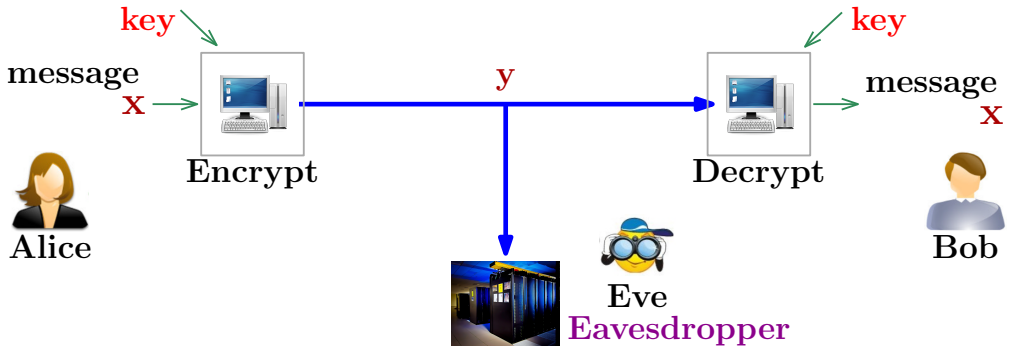
If a randomly chosen 200-digit number $n$ is tested for $\approx 100$ different values of $a$ without getting a Fermat witness, then almost surely $n$ is prime

# Private Key Cryptography

Alice sends message to Bob, Eve eavesdrops



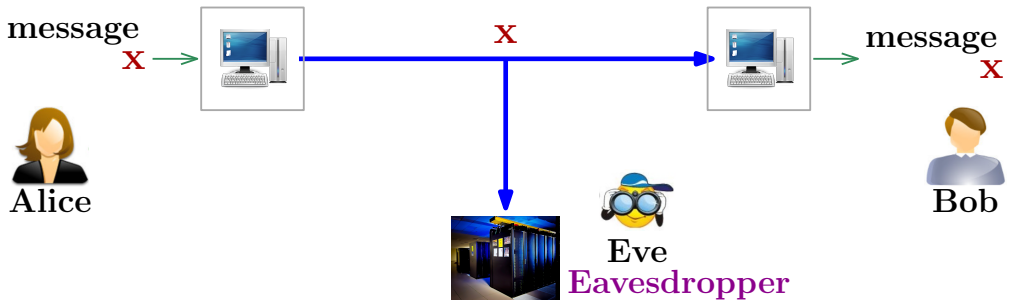

Exchange the encryption key for a good cipher!



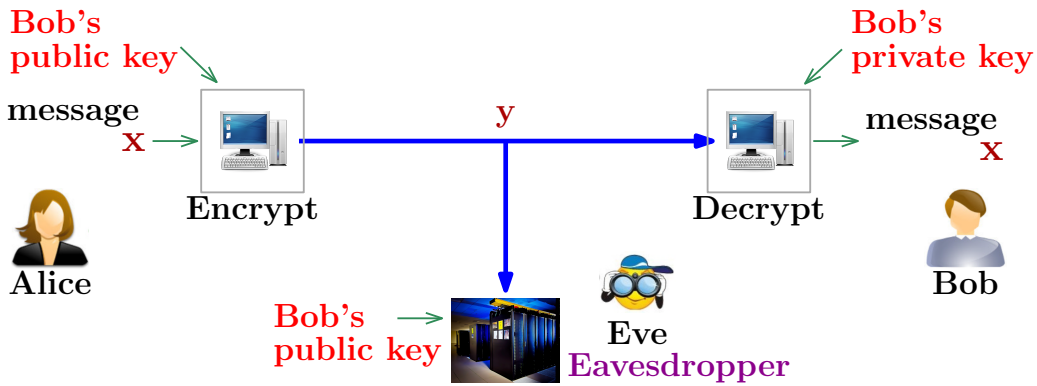

But during **key exchange**, Eve could get the key and all security is lost!

# Public Key Cryptography

Alice sends message to Bob, Eve eavesdrops




Everyone knows public key, only Bob knows private key




Alice encrypts with public key, Bob decrypts with private key

Eve cannot do anything! ▷ No key exchange

# Public Key Cryptography: RSA

## Keys generation

- Choose two large primes $p$ and $q$ ▷ $p$ and $q$ are secret
- Set $n = pq$ and $T = (p-1)(q-1)$
- Choose $e$ such that $\text{GCD}(e, T) = 1$ ▷ $e^{-1}$ exists
- choose $d = e^{-1}$ modulo $T$ ▷ $de \equiv_T 1$
- $e$ and $n$ are public keys ▷ published on Internet
- $d$ is private key ▷ only Bob knows it

## Encryption

- Encode message as an integer $M < n$
- Compute $C = M^e \ \% \ n$ ▷ Use modular exponentiation!

## Decryption

- Compute $M = C^d \ \% \ n$ ▷ Use modular exponentiation!

# Public Key Cryptography: RSA

## Keys generation

- Choose two large primes $p$ and $q$
- Set $n = pq$ and $T = (p-1)(q-1)$
- Choose $e$ such that $\text{GCD}(e, T) = 1$
- choose $d = e^{-1}$ modulo $T$
- $e$ and $n$ are public keys
- $d$ is private key

## Example Keys

- $p = 59$ and $q = 43$
- $n = 2537$ and $T = 2436$
- $e = 13$: $\text{GCD}(13, 2436) = 1$
- $d = 937 = 13^{-1}$ modulo $T$
- 13 and 2537 are public keys
- 937 is private key

Encrypt "STOP" $\quad S \to 18, T \to 19, O \to 14, P \to 15 \implies 1819\ 1415$

$C = M^e \ \%\ n \qquad 1819^{13} \ \%\ 2537 = 2081 \qquad 1415^{13} \ \%\ 2537 = 2182$

Encrypted message is 2081 2182

Decrypt "0981 0461"

$M = C^d \ \%\ T \qquad 0981^{937} \ \%\ 2537 = 0704 \qquad 0461^{937} \ \%\ 2537 = 1115$

$07 \to H, 04 \to E, 11 \to L, 15 \to P \implies$ "HELP" $\quad \triangleright$ message is "HELP"

# RSA: Correctness and Security

## Encryption

- Encode message as an integer $M < n$
- Compute $C = M^e \% n$ ▷ Use modular exponentiation!

## Decryption

- Compute $M = C^d \% n$ ▷ Use modular exponentiation!

We need to show that

- $C^d \% n$ is indeed equal to $M$ ▷ Correctness
- Without knowing $d$ cannot compute $M$ from $C$ ▷ Security

# RSA: Proof of Correctness

The Correctness of RSA relies on the fact that the encryption and decryption processes are inverses of each other

**Correctness of RSA:** $C^d = (M^e)^d \equiv_n M$

**Proof:** $de \equiv_T 1$ Thus, $\exists\, k \in \mathbb{Z} : de = 1 + k(p-1)(q-1)$. So

$C^d = M^{de} \equiv_{pq} M^{1+k(p-1)(q-1)}$

- $C^d = M(M^{p-1})^{k(q-1)} \equiv_p M \cdot 1^{k(q-1)} \equiv_p M$
- $C^d = M(M^{q-1})^{k(p-1)} \equiv_q M \cdot 1^{k(p-1)} \equiv_q M$ ▷ FLT

Hmm! a system of modular equations with $\text{GCD}(p, q) = 1$

$C^d \equiv_{pq} M$ is a solution to this system and by CRT its a unique solution

# RSA: Proof of Security

Without knowing $d$ cannot compute $M$ from $C$ ▷ Security

**Hardness of the Factorization Problem**

To break RSA, an attacker would need to factor the modulus $n = p \times q$. It is believed to be very hard to find $p$ and $q$ given $n = pq$

Prime factorization is a difficult problem

▷ though we do not have theoretical proof for it

- Key Generation: The difficulty of deriving $p$ and $q$ from $n$ ensures that the private key remains secure.
- Public Key Exposure: The public key $(e, n)$ is shared openly, but it is infeasible to compute the corresponding private key d without knowledge of $p$ and $q$
- One-Way Function: RSA's encryption and decryption are based on one-way mathematical functions that are easy to compute in one direction but hard to reverse

# Challenges to RSA Security

Challenges that can compromise RSA security:

- Small Key Sizes: Using smaller key sizes (e.g., 512-bit keys) makes RSA susceptible to brute-force attacks. Modern implementations require much larger key sizes (e.g., 2048-bit or 4096-bit) to ensure security
- Computational Complexity: RSA operations are computationally intensive, especially as the key size increases. This requires efficient algorithms and hardware to handle large numbers
- Quantum Computing: Quantum algorithms (theoretically) break RSA encryption by efficiently factoring large numbers
- Side-Channel Attacks: RSA implementations can be vulnerable to side-channel attacks (e.g., timing attacks) that exploit weaknesses in the algorithm's execution or hardware

# Mitigating RSA Vulnerabilities

There are several strategies to mitigate vulnerabilities in RSA encryption:

- Increasing Key Sizes: Using larger key sizes (e.g., 2048-bit or 4096-bit) to make factorization infeasible for classical computers
- Hybrid Cryptosystems: Combining RSA with symmetric encryption algorithms (e.g., AES) to reduce computational overhead while maintaining security
- Quantum-Resistant Algorithms: Researchers are focusing on developing algorithms that can withstand quantum attacks, ensuring long-term security for encrypted communications

# Hybrid Cryptosystems and RSA

Hybrid cryptosystems combine asymmetric (e.g., RSA) & symmetric encryption (e.g., AES) to balance between security and performance

- Key Exchange with RSA: RSA is used to encrypt and securely exchange a symmetric key (e.g., AES key) between sender and receiver
- Data Encryption with AES: Once the symmetric key is securely exchanged, AES is used to encrypt large data efficiently
- Decryption Process: The recipient uses RSA to decrypt the symmetric key and then uses it to decrypt the data encrypted with AES

- Enhance performance: RSA provides strong security, but it is computationally intensive – Symmetric encryption (e.g., AES) are much faster but rely on the secure key exchange
- Widely Used in Practice: Commonly used in SSL/TLS protocols (e.g., HTTPS) for secure web browsing and in email encryption systems

# Quantum Computing and RSA's Vulnerabilities

RSA (Rivest-Shamir-Adleman) has been a cornerstone in securing online transactions for decades. However, the rise of quantum computing poses a threat to RSA's security:

- Quantum Computing: Quantum Computing promises to revolutionize computing by enabling much faster computation than classical computers
- Shor's Algorithm: A quantum algorithm that can break RSA encryption by factoring large numbers in polynomial time ($O((\log N)^3)$)

  ▷ With enough quantum processing power, RSA's reliance on the difficulty of factoring large primes would no longer be secure

  ▷ Quantum Supremacy: Quantum computers capable of breaking RSA are still in the experimental phase, but the potential is real, and tech companies are preparing for this eventuality
- Post-Quantum Cryptography: Algorithms that are resistant to quantum attacks
  - Lattice-based cryptography
  - Code-based cryptography
  - Multivariate-quadratic-equations-based cryptography

Transition to Post-Quantum Cryptography: Online payment systems will transition to PQC before commercial quantum computers are available

# Why Post-Quantum Methods Are Quantum-Safe

Unlike RSA or ECC, post-quantum cryptographic schemes are built on mathematical problems that are hard for both classical and quantum computers:

- Lattice-Based Cryptography:
  - Based on problems like Shortest Vector Problem (SVP) and Learning With Errors (LWE)
  - These problems remain hard even for quantum algorithms—no known quantum polynomial-time algorithm exists for solving them
- Code-Based Cryptography:
  - Relies on the hardness of decoding general linear error-correcting codes
  - The McEliece cryptosystem (1978) still resists all known classical and quantum attacks
- Multivariate Polynomial Cryptography:
  - Based on the difficulty of solving systems of nonlinear multivariate equations over finite fields
  - Solving these systems (MQ problem) is NP-HARD and quantum algorithms don't offer speedups here

# Post-Quantum Cryptography (PQC)

Post-Quantum Cryptography (PQC) refers to cryptographic algorithms that are secure against quantum computing threats:

- Quantum-Safe Algorithms: Algorithms like lattice-based cryptography, hash-based cryptography, and code-based cryptography are being researched as alternatives to RSA
- NIST's PQC Standardization: The National Institute of Standards and Technology (NIST) is leading an effort to standardize post-quantum cryptographic algorithms that can replace RSA and other classical systems.
- Hybrid Approaches: Some systems are being designed to combine quantum-resistant algorithms with RSA to create quantum-safe hybrid cryptosystems.