Information Security

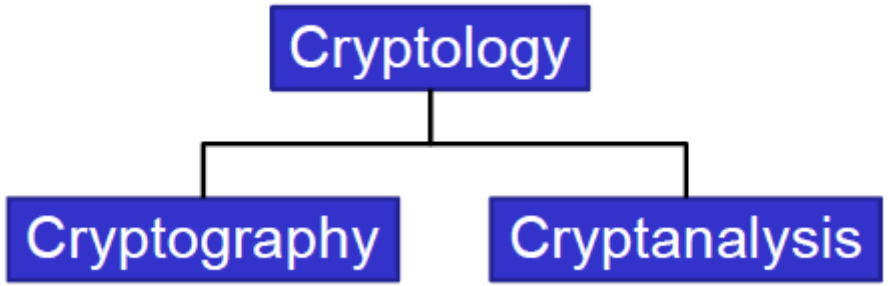
Lecture 2: Cryptography

Mona Taghavi

LaSalle College
Montréal

Terminology

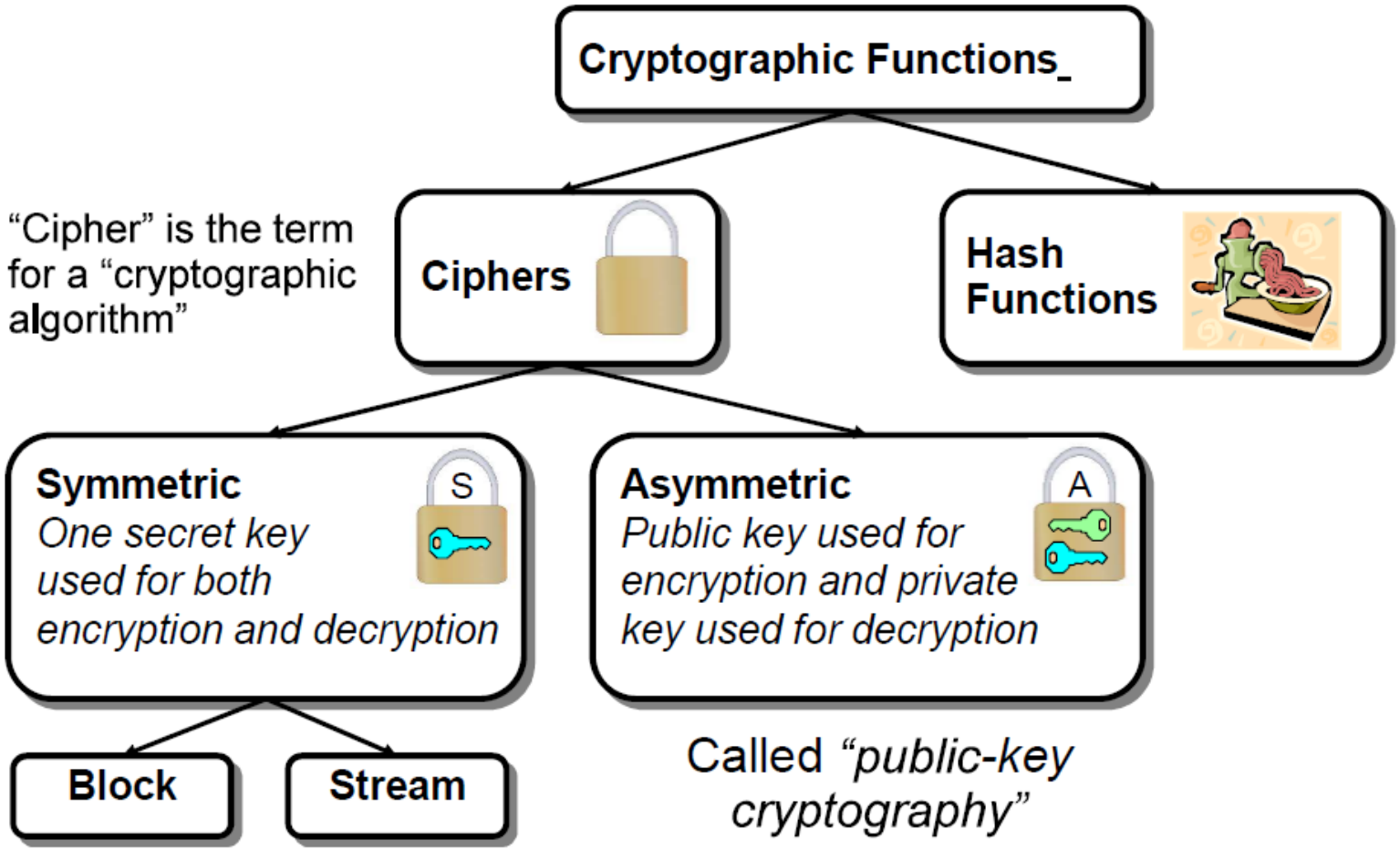
- **Greek Word**: “Krypto” = Hidden / Secret
- Cryptography is the science of secret writing with the goal of hiding the meaning of a message.
- Cryptanalysis is the science of breaking cryptography.
- Cryptology covers both cryptography and cryptanalysis.



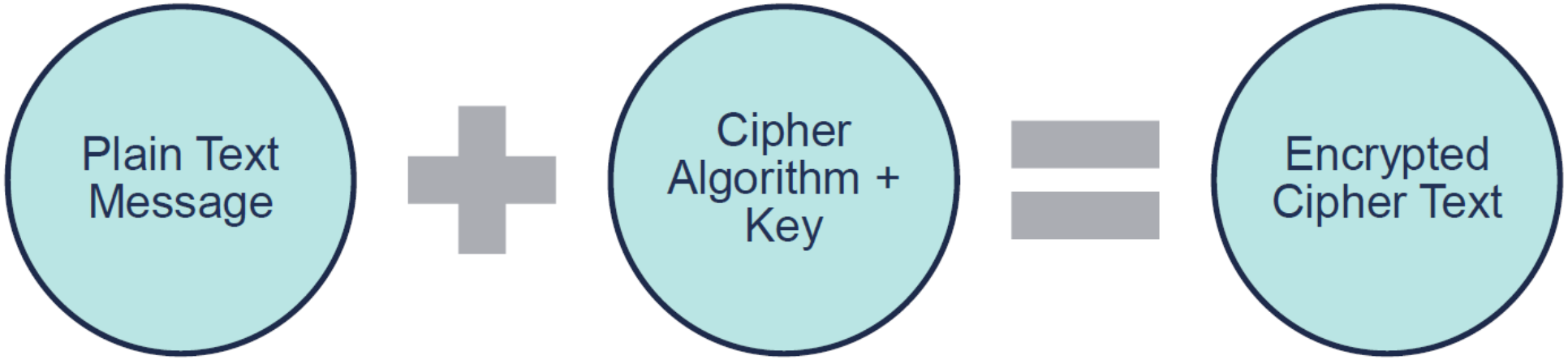
What can cryptography do?

- Crypto can provide the following security services:
- Confidentiality:
  - Makes data unreadable to entities who do not have the appropriate cryptographic keys, even if they have the data.
- Data Integrity:
  - Entities with the appropriate cryptographic keys can verify that data is correct and has not been altered, either deliberately or accidentally.
- Authentication:
  - Entities who communicate can be assured that the other user/entity or the sender of a message is what it claims to be.
- Digital Signature and PKI (Public-Key Infrastructure):
  - Strong proof of data origin which can be verified by 3rd parties.
  - Scalable (to the whole Internet) distribution of cryptographic keys.

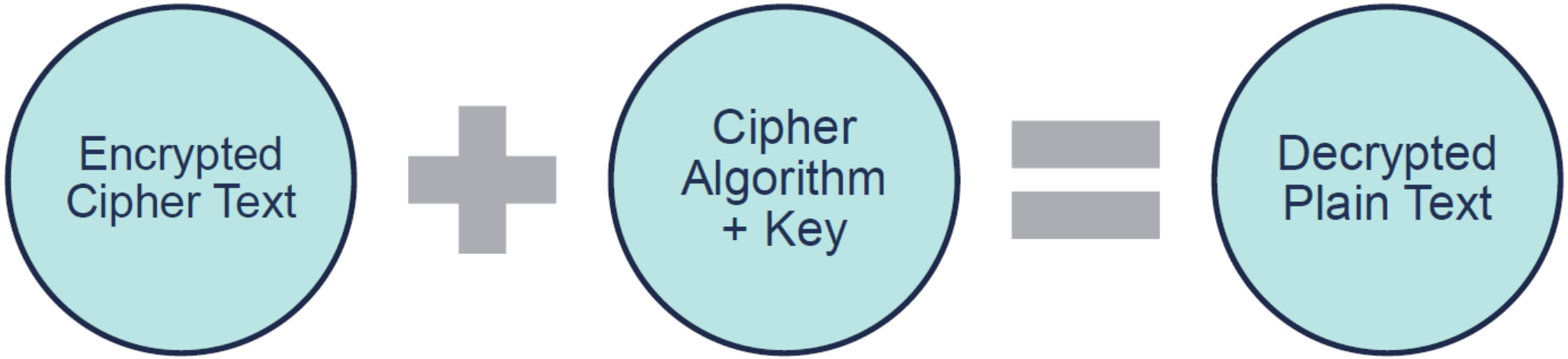
Taxonomy of cryptographic functions



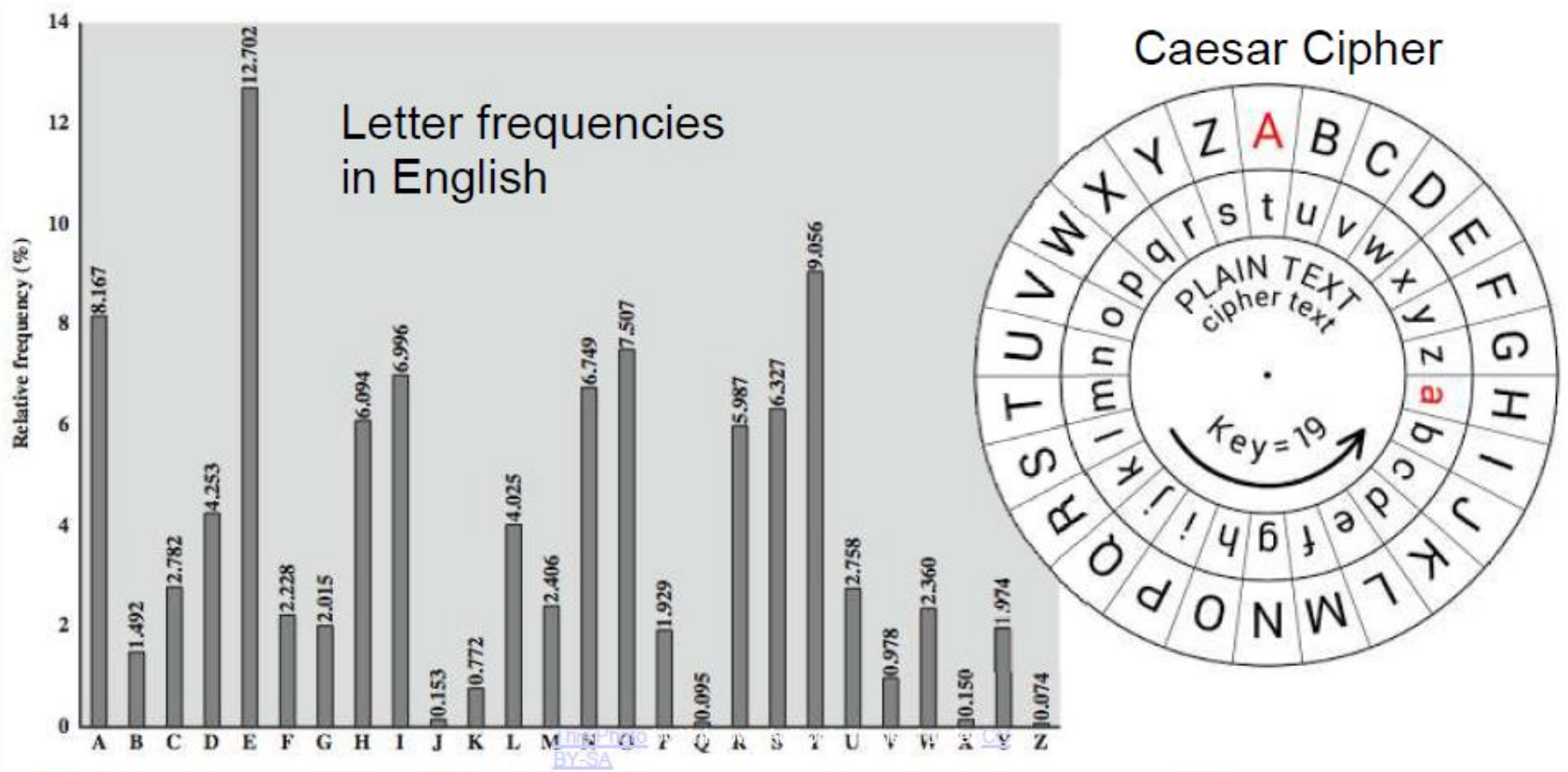
Encrypting a Message



Decrypting a Message

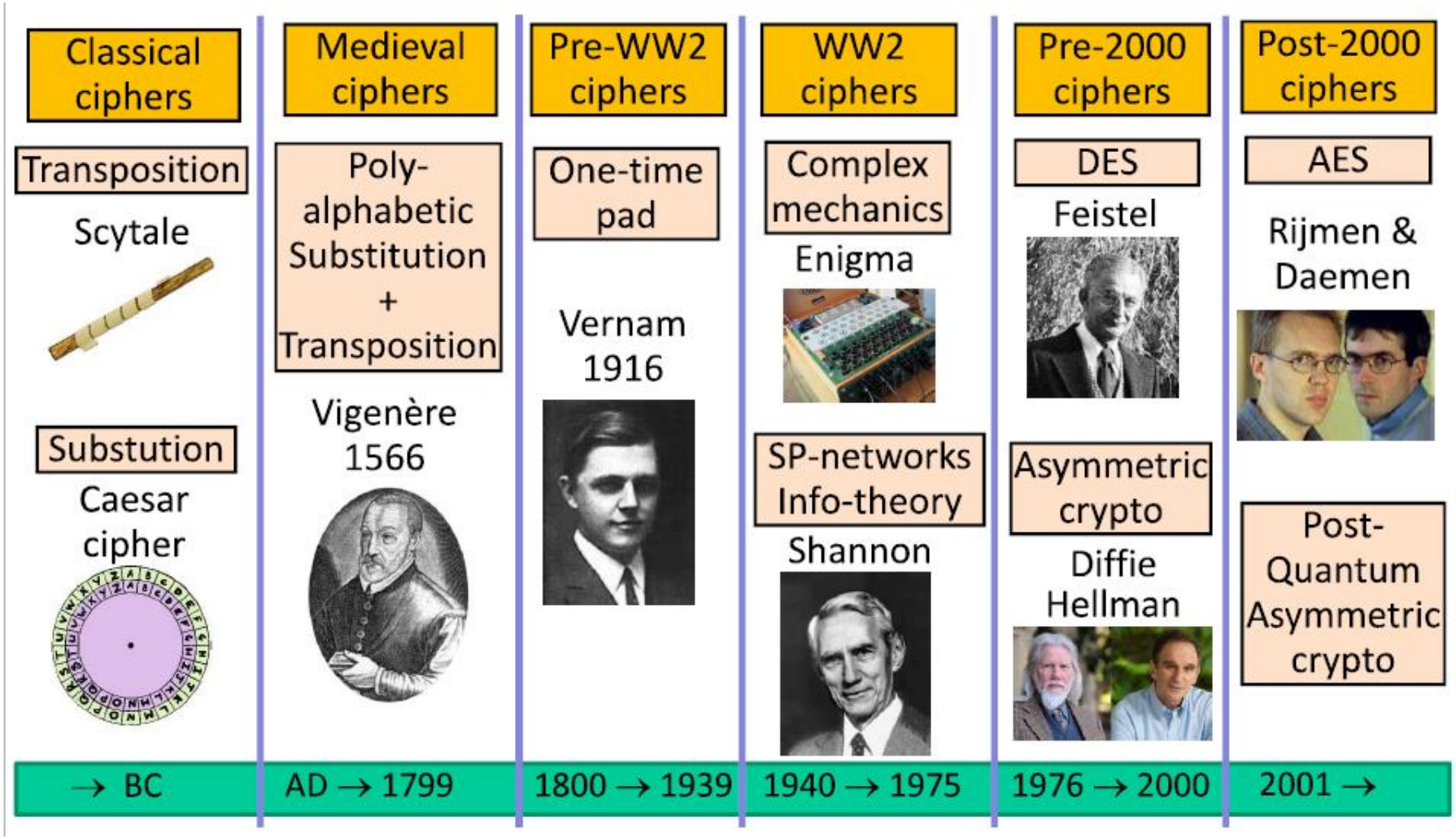


Letter Frequencies



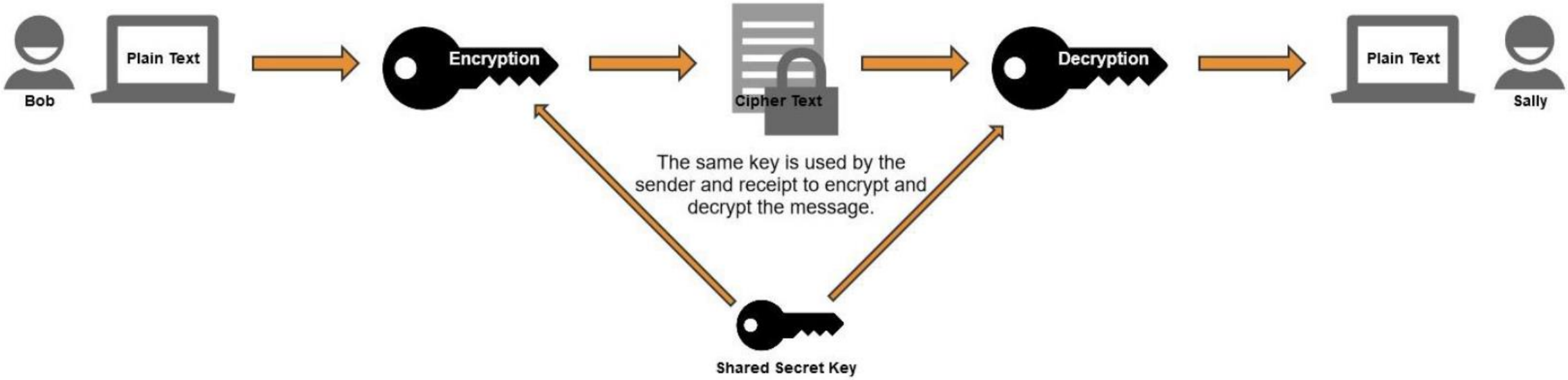
Historic ciphers, like the Caesar Cipher, are weak because they fail to hide statistical regularities in the ciphertext.

Evolution of Ciphers



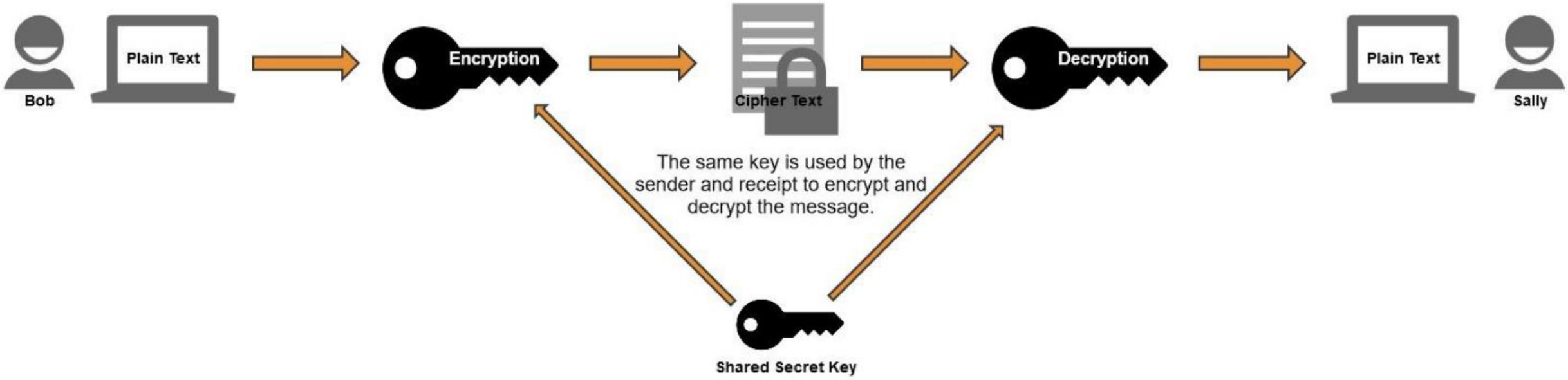
Symmetric (Private Key) Encryption

- Symmetric encryption uses a **single key** for **encryption** and **decryption**.
- Both the **sende**r and **receiver** have the **same key** and use it to encrypt and decrypt all messages.
- It's also known as **secret-key encryption** or **private-key** encryption.



Symmetric (Private Key) Encryption

- **Symmetric encryption** is much more efficient at encrypting large amounts of data than its counterpart, **asymmetric encryption**.
- The downside of symmetrical encryption is that it makes it hard to initiate communication the first time.



Symmetric Encryption Algorithm: DES & 3DES

- **Data Encryption Standard (DES)**
- DES is an older algorithm that widely used for a period of time dating back to the 1970’s.
- It has been compromised and no longer secure.

- **Triple DES (3DES)**
- 3DES was developed as an improvement over DES.
- It improved the encryption by encrypting the data with DES three times with two, or sometimes three keys.
- While 3DES is a significant improvement over DES, it consumes a lot of processer power and memory resources.
- AES is much less resource-intensive and has replaced 3DES as the current standard.

Symmetric Encryption Algorithm: AES

- **Advanced Encryption Standard (AES)**
- AES is a very strong encryption algorithm that's commonly used worldwide.
- It's significantly faster than both DES and 3DES and also provides stronger encryption.
  - 128-Bit AES would take billion of years to brute force
- It's also the "official" encryption standard for the U.S. government (since 2002).
- Governments, militaries, banks and corporations rely on it. It's responsible for securing most, if not all of your personal and financial data. There are special CPU instructions for it.

AES Algorithm (*Rijndael)*

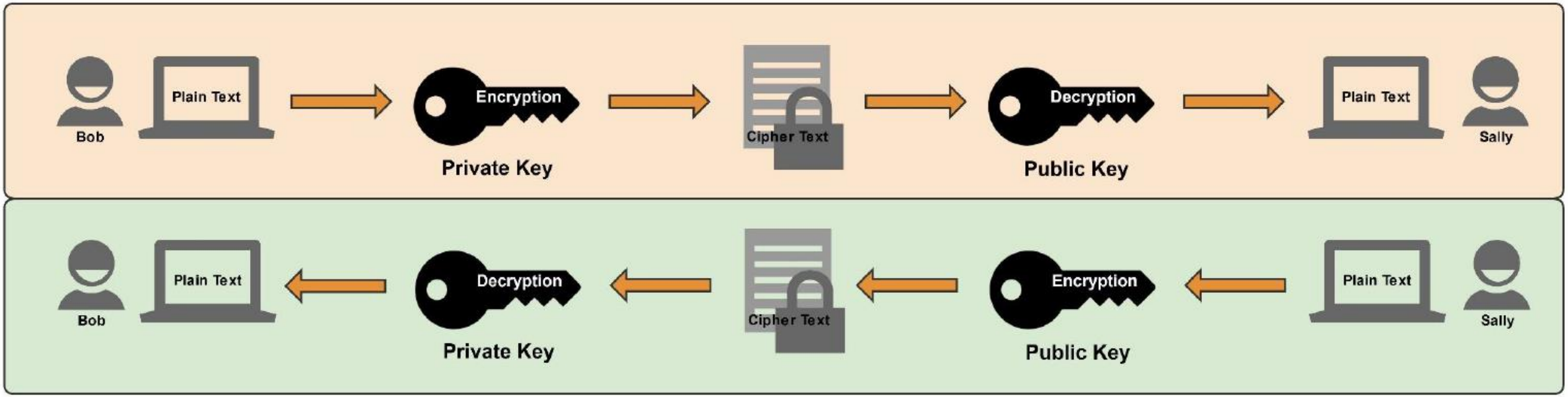
- AES can be performed with the following key sizes: **128 bits, 196 bits and 256 bits**. Generally, increasing the key size also increases the level of security.
- It is also a *block cipher*, meaning that the data is divided into blocks before encryption. AES divides plaintext into blocks of 16 bytes (128 bits).
- we arrange each block of the plaintext into a 4x4 matrix and repeatedly perform a set of operations on it. We call each iteration a *round*, and we perform **10, 12 or 14 rounds** depending on the key length:
  - 10 rounds for a 128-bit key
  - 12 rounds for a 196-bit key
  - 14 rounds for a 256 bit key
- For each round, we generate a *round key* from the main key using the *Rijndael Key Schedule*.

Asymmetric Encryption

- Asymmetric encryption uses two keys, a public key and a private key created as a matched pair.
  - **Private Key**: Kept secret and never shared.
  - **Public Key**: Shared with others.
- Commonly referred to as:
  - Public Key Encryption
  - Public Key Infrastructure (PKI) Encryption

How Public Key Encryption Works

- Anything encrypted with the **private key** can only be decrypted with the matched **public key**.
- Anything encrypted with the **public key** can only be decrypted with the matched **private key**.



Diffie-Hellman

- Diffie-Hellman key exchange is a method of digital encryption that securely exchanges cryptographic keys between two parties over a public channel without their conversation being transmitted over the internet.
- The two parties use symmetric cryptography to encrypt and decrypt their messages.

Agree upon two numbers:

P Prime Number 13

G Generator of P 6

Private = 5

Randomly generate a Private Key

Private = 4

(6^{5}) MOD 13

(7776) MOD 13

Public = 2

Calculate Public Key:

$(G^{Private})$ MOD P

(6^{4}) MOD 13

(1296) MOD 13

Public = 9

Exchange Public Keys

(9^{5}) MOD 13

(59049) MOD 13

Shared Secret = 3

Calculate the Shared Secret

$(Shared\ Public^{Private})$ MOD P

(2^{4}) MOD 13

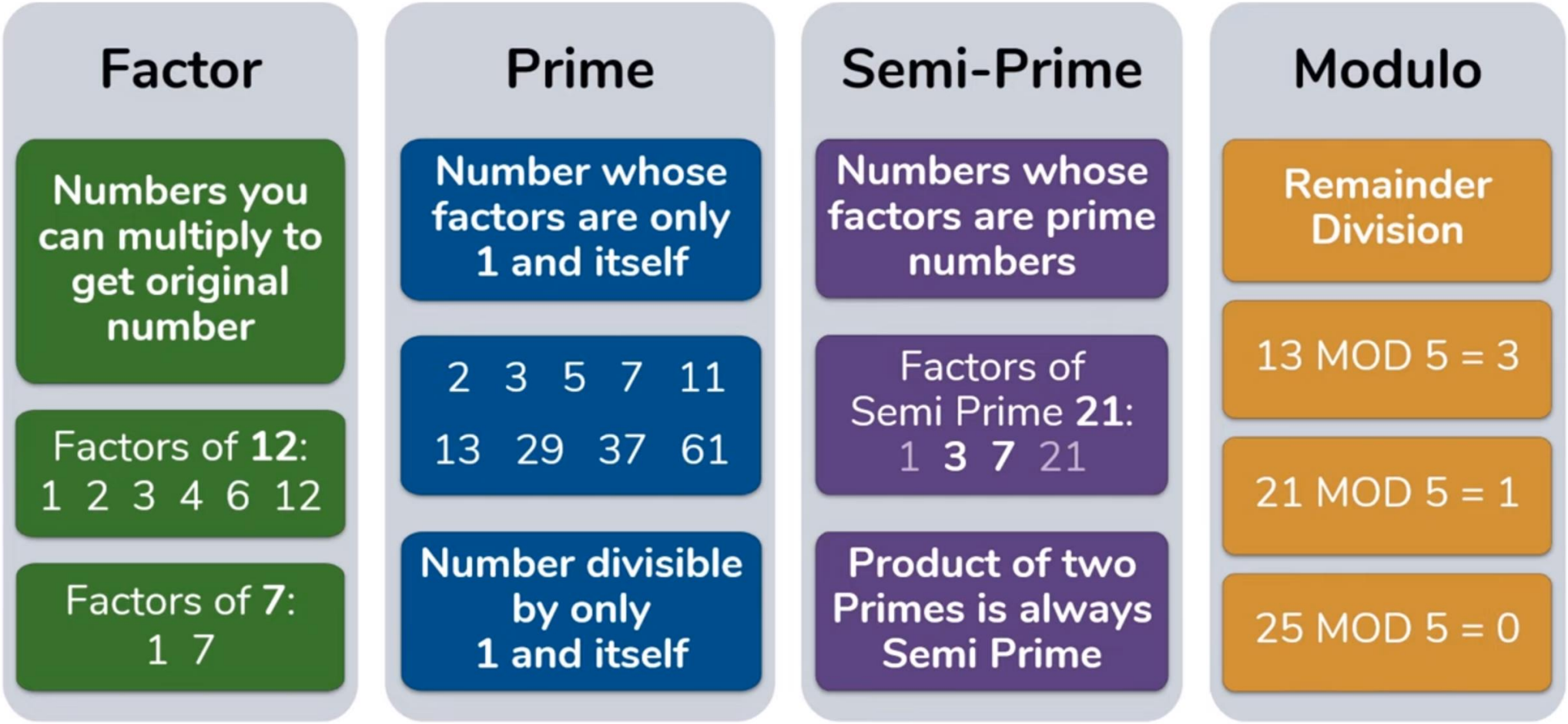
(16) MOD 13

Shared Secret = 3

THE RIVEST-SHAMIR-ADLEMAN (RSA) ALGORITHM FOR PUBLIC-KEY CRYPTOGRAPHY — THE BASIC IDEA

- It is named after its inventors Ron Rivest, Adi Shamir and Len Adleman.
- Published in 1978
- It is the most widely used public-key encryption algorithm today.
- It provides confidentiality and digital signatures.
- Its security is based on the difficulty of integer factorization

Basic math concepts



RSA Example

- Generating Keys:
  - Select two Prime Numbers (P,Q)
  - Calculate Product (P*Q)
  - Calculate Totient (P-1)*(Q-1)
  - Select Public Key (E)
    - Must be Prime
    - Must be less than Totient
    - Must NOT be a factor of the Totient
  - Select a Private Key (D)
    - Product of D and E, divided by T must result in a remainder of 1
    - (D*E) MOD T = 1

Prime #s	P Q	7 19
Product	N	133
Totient	T	108
Public Key	E	29
Private Key	D	41

RSA Encryption and Decryption

- Encryption and Decryption:
  - Encryption:

    $Message^E \bmod N = Cipher\ Text$
  - Decryption:

    $Cipher^D \bmod N = Message$

- Encryption and Decryption:
  - Encryption:

    $Message^D \bmod N = Cipher\ Text$
  - Decryption:

    $Cipher^E \bmod N = Message$

RSA Example

Message
60

Prime #s	P Q	7 19
Product	N	133
Totient	T	108
Public Key	E	29
Private Key	D	41

- Encryption and Decryption:
  - Encryption:

    Message^E MOD N = *Cipher Text*
  - Decryption:

    Cipher^D MOD N = *Message*

- **Encrypt** with **Public Key**, **Decrypt** with **Private Key**

(60^29) MOD 133 = 86

(86^41) MOD 133 = 60

RSA Example

Message
60

Prime #s	P Q	7 19
Product	N	133
Totient	T	108
Public Key	E	29
Private Key	D	41

- Encryption and Decryption:
  - Encryption:

    Message^D MOD N = *Cipher Text*
  - Decryption:

    Cipher^E MOD N = *Message*

- **Encrypt** with **Private Key**, **Decrypt** with **Public Key**

(60^41) MOD 133 = 72

(72^29) MOD 133 = 60

RSA

- How secure is RSA?
  - Security lies in difficulty of factoring Semi Prime numbers
    - If given the number 133, could you extract 7 and 19?
      - Really? … Prove it, what are the factors of 1909?
  - In 1991, RSA Laboratory created the RSA Challenge:
    - Released 54 Semi-Primes of various sizes and asked for Factors
      - Competition ended in 2007 – only 12 factors were identified
      - As of 2020, another 11 were identified (no cash was awarded)
        - Biggest number factored: 829 Bits (Feb 2020)
      - In 29 years, the 1024 bit number has never been factored

RSA

- How secure is RSA?

1024 bit Semi-Prime number:

```
135066410865995223349603216278805969938881475605667027524485143851526510604859533833940287150571909441798207282164471551373680419703964191743046496589274256239341020864383202110372958725762358509643110564073501508187510676594629205563685529475213500852879416377328533906109750544334999811150056977236890927563
```

- 1024 bit RSA keys was recommended standard since 2002
- 2048 bit RSA Keys is recommended standard since 2015