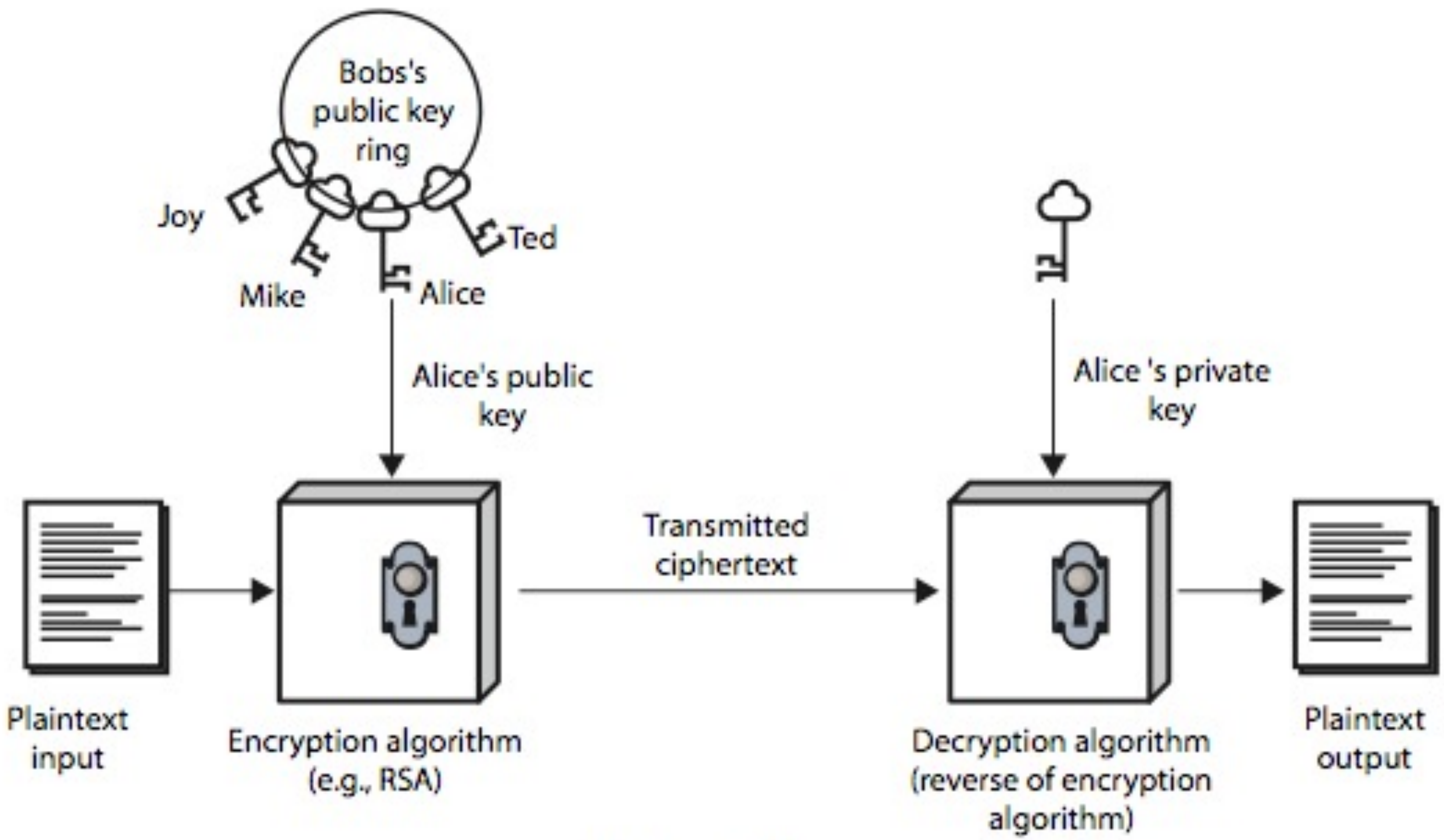
Why Public-Key Cryptography?

- Developed to address two key issues:
  - **key distribution** – how to have secure communications in general without having to trust a key distribution center with your key
  - **Digital signatures** – how to verify a message comes intact from the claimed sender
- Public invention due to Diffie & Hellman at Stanford University in 1976
  - known earlier in classified community

Public-Key Cryptography

- **Public-key/two-key/asymmetric** cryptography involves the use of **two** keys:
  - a **public-key**, which may be known by anybody, and can be used to **encrypt messages**, and **verify signatures**
  - a **private-key**, known only to one party, used to **decrypt messages**, and **sign** (create) **signatures**
- **Asymmetric** because
  - those who encrypt messages or verify signatures **cannot** decrypt messages or create signatures

Public-Key Cryptography



(a) Encryption

Public-Key Characteristics

- Public-Key algorithms rely on two keys where:
  - when the relevant (en/decrypt) key is known it is computationally easy to en/decrypt messages
  - it is computationally infeasible to find decryption key, knowing only algorithm & encryption key

Diffie & Hellman Key Exchange

- Alice and Bob want to share a secret (e.g., a key) in an open channel.

- Assume they agree on two numbers n and g
- g is primitive root mod (n)
  - For each $p < n$ s.t. p is coprime to n, there is an a such that

    $g^a = p \bmod (n)$
- These g and n do not have to be kept secret

Alice

- Chooses a large random number x
- Calculates

$$X = g^x \bmod (n)$$

- Sends X, g, and n to Bob.

Bob

- Chooses a large random number y
- Calculates

$$Y = g^y \bmod (n)$$

- Sends Y to Alice.

- Alice calculates

$$k = Y^x \bmod (n)$$

- Bob calculates

$$k' = X^y \bmod (n)$$

The Key

- $k' = k$ is the shared key

$$k = Y^x \bmod (n) = (g^y)^x \bmod (n) = g^{yx} \bmod (n)$$

$$k' = X^y \bmod (n) = (g^x)^y \bmod (n) = g^{xy} \bmod (n)$$

No efficient classical algorithm for computing general discrete logarithms.

- Nobody can calculate k given

n, g, X, and Y

Example

- Alice and Bob get public numbers
  - $n = 23$, $g = 9$
- Alice and Bob compute public values
  - $X = 9^4 \bmod 23 = 6561 \bmod 23 = 6$
  - $Y = 9^3 \bmod 23 = 729 \bmod 23 = 16$
- Alice and Bob exchange public numbers

Example

- Alice and Bob compute symmetric keys
  - $k_a = 16^4 \bmod 23 = 9$
  - $k_b = 6^3 \bmod 23 = 9$
- Alice and Bob now can talk securely!

Diffie-Hellman Limitations

- Only Alice and Bob know k
- Good for only one session
- Used if you only want a symmetric key
- Can’t be sure connected to the same person
- No authentication

RSA

- By Rivest, Shamir & Adleman of MIT in 1977
- Best known & widely used public-key scheme
- Uses large integers (e.g., 1024 bits)
- Security due to cost of factoring large numbers, and difficulty of computing discrete logarithm.

Background

- Totient function $\phi(n)$
  - Number of positive integers less than *n and relatively prime to n*
  - *Relatively prime means with no factors in common with n*
- Example: $\phi(10) = 4$
  - 1, 3, 7, 9 are relatively prime to 10
- If $n=pq$, $\phi(n) = (p-1)(q-1)$

- Euler's Theorem: $a^{\phi(n)} \bmod n = 1$ where $\gcd(a,n)=1$
  - Example: $n=10$, $a=3$.

RSA Key Setup

- Each user generates a public/private key pair by:
- Selecting two large primes at random p, q
- Computing their system modulus $n=p*q$

  $\phi(n)=(p-1)(q-1)$
- Selecting at random the encryption key e

  where $1<e<\phi(n)$, $\gcd(e,\phi(n))=1$
- Solve following equation to find decryption key d

  $e*d=1 \bmod \phi(n)$ and $0\le d\le n$
- Publish their public encryption key: PU={e,n}
- Keep secret private decryption key: PR={d,n}

p and q are kept secrete.

RSA Use

- To encrypt a message M, the sender:
  - obtains **public key** of recipient `PU={e,n}`
  - computes: $C = M^e \bmod n$, where $0 \leq M < n$
- To decrypt the ciphertext C the owner:
  - uses their private key `PR={d,n}`
  - computes: $M = C^d \bmod n$
- Note that the message M must be smaller than the modulus n (block if needed)