ROSE-HULMAN
INSTITUTE OF TECHNOLOGY

CSSE 490
Network Security

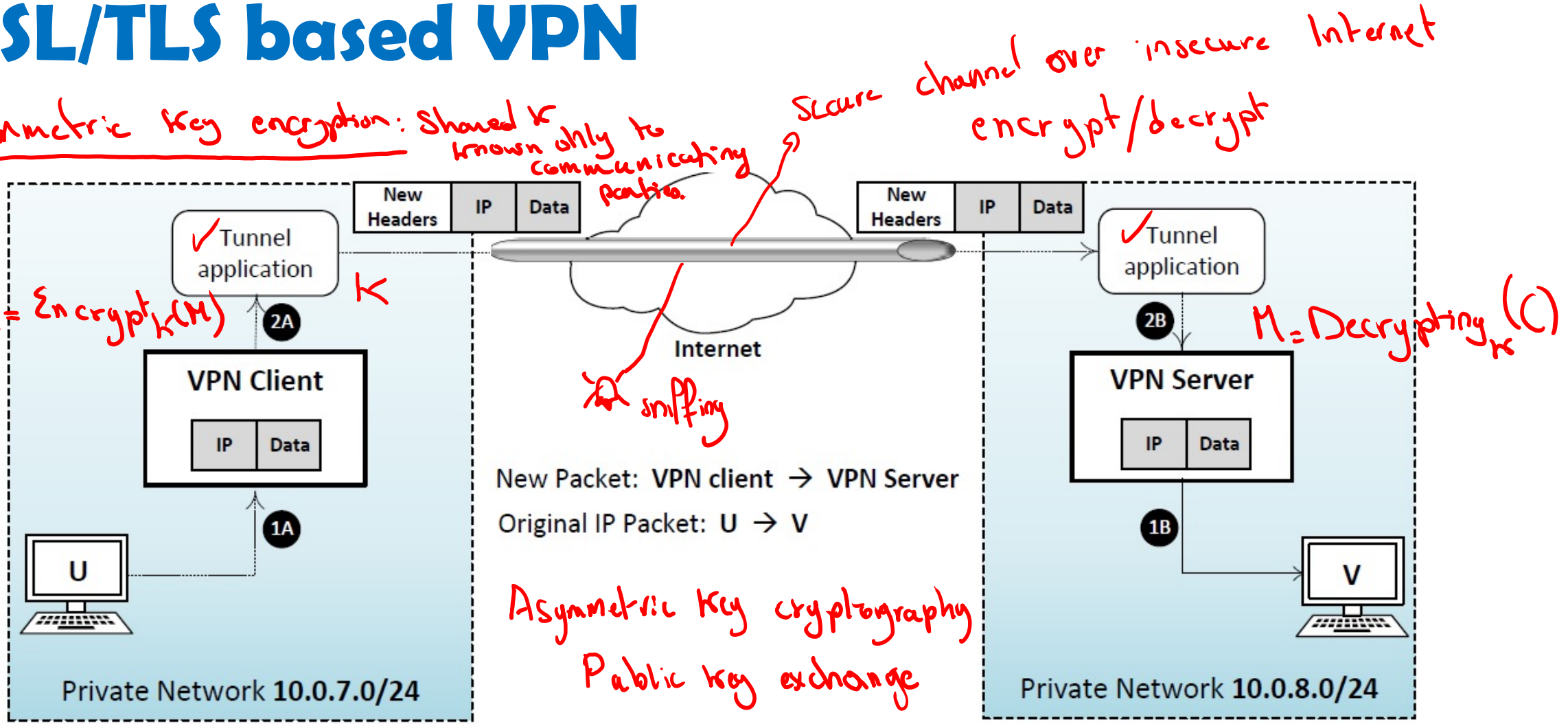
Day 30: Public-Key Encryption

Slides modified from slides designed by Professor Kevin (Wenliang) Du

Outline

- Recap: SSL/TLS-based VPN
- A bit of history
- Public-key cryptography
- Diffie-Hellman key exchange
- Rivate-Shamir-Adleman (RSA) math background
- RSA algorithm

SSL/TLS based VPN



A bit of history

Whitfield Diffie Martin Hellman

2015 Turing Award Winner

Clifford Cocks

Leonard Adleman Ron Rivest Adi Shamir

2002 Turing Award Winner

Public-key cryptography

Generate two sets of keys

- Public key for encryption
- Private key for decryption

$C = Encrypt_{publickey}(M)$ → (insecure channel)

$M = Decrypting_{private\ key}(C)$

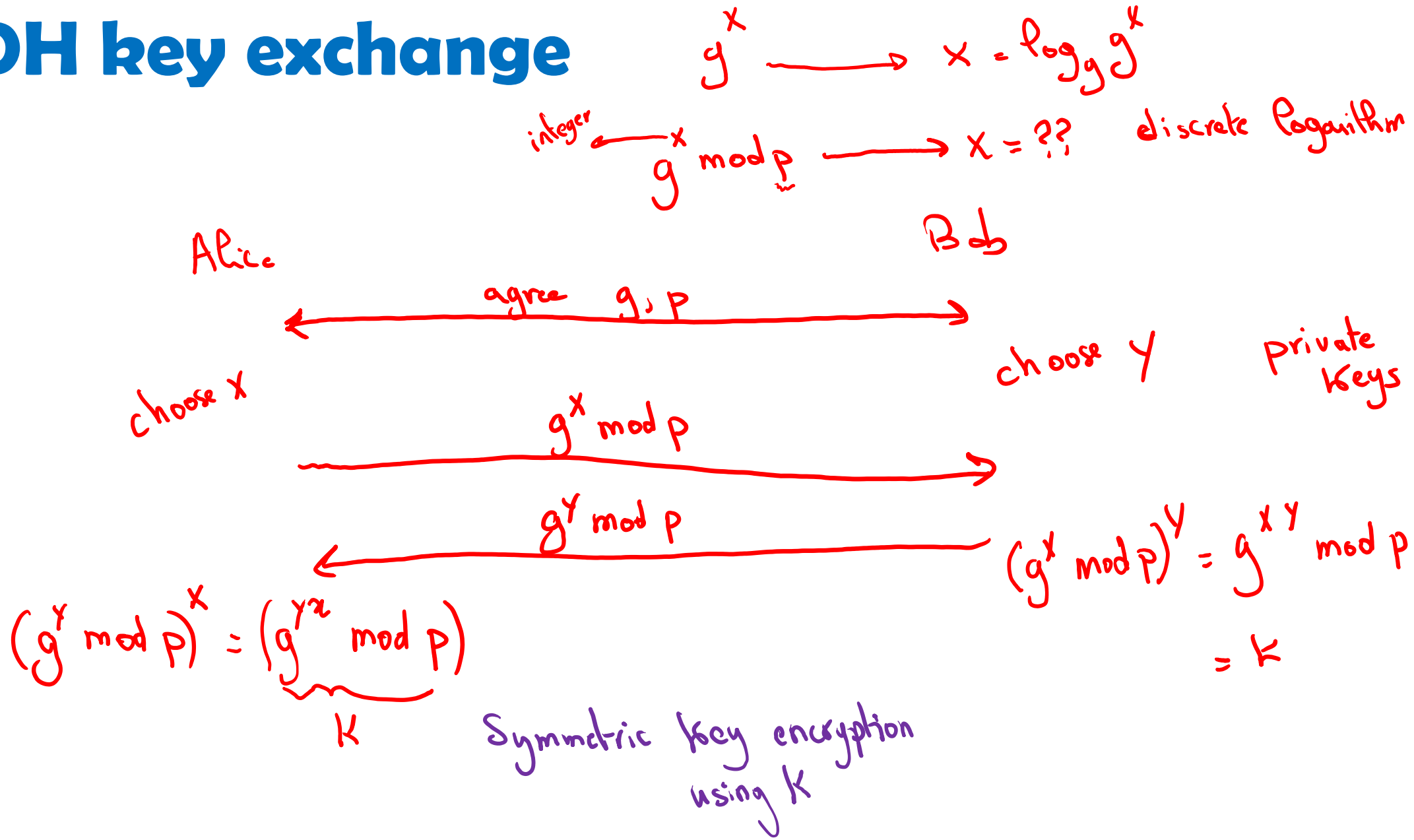
Also, for authentication

- Private key for signature
- Public key for verifying the signature

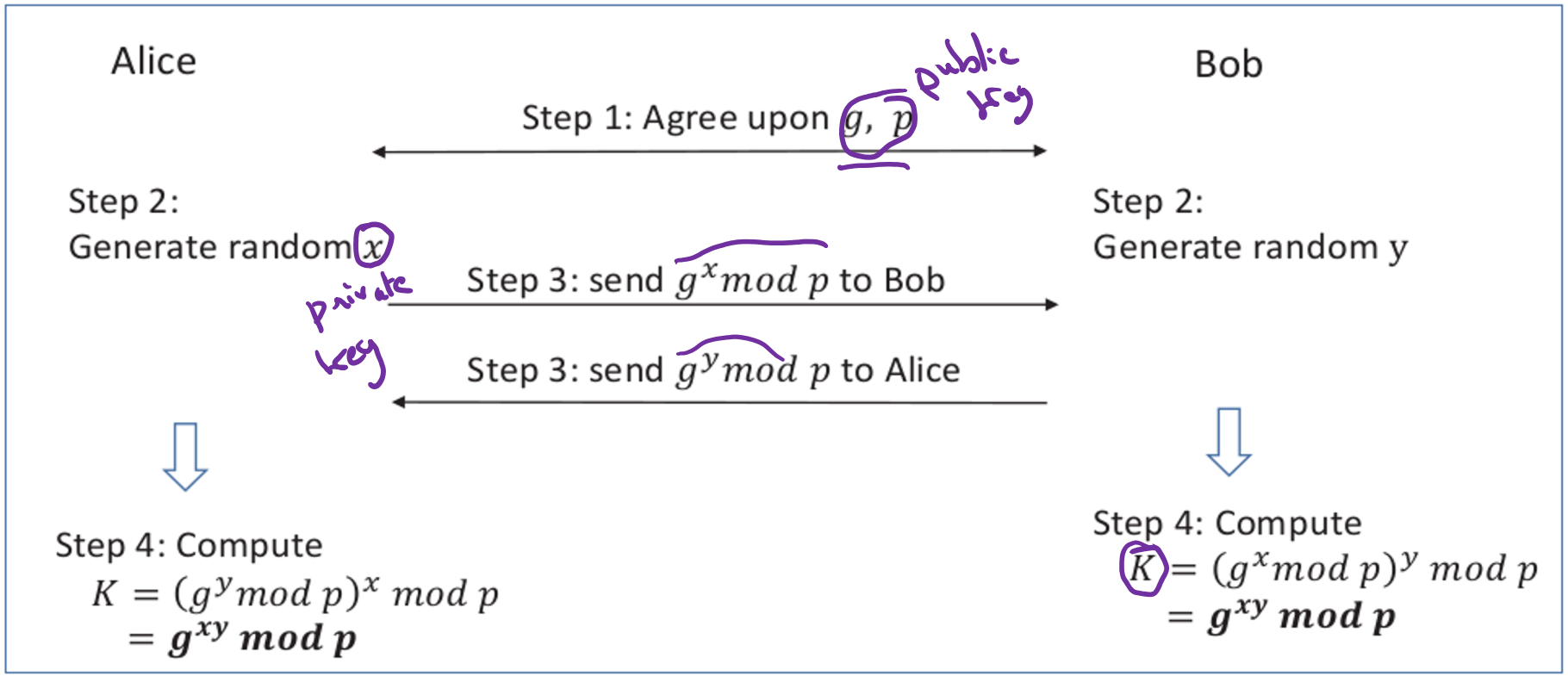
Diffie-Hellman key exchange

- Exchange secret key over insecure channel
- Communicating parties agree on:
  - Number p: big prime number (2048-bit number)
  - Generator g: small prime number
- Alice picks x, a random positive integer, x < p
- Bob pick y, a random positive integer, y < p

DH key exchange



DH key exchange

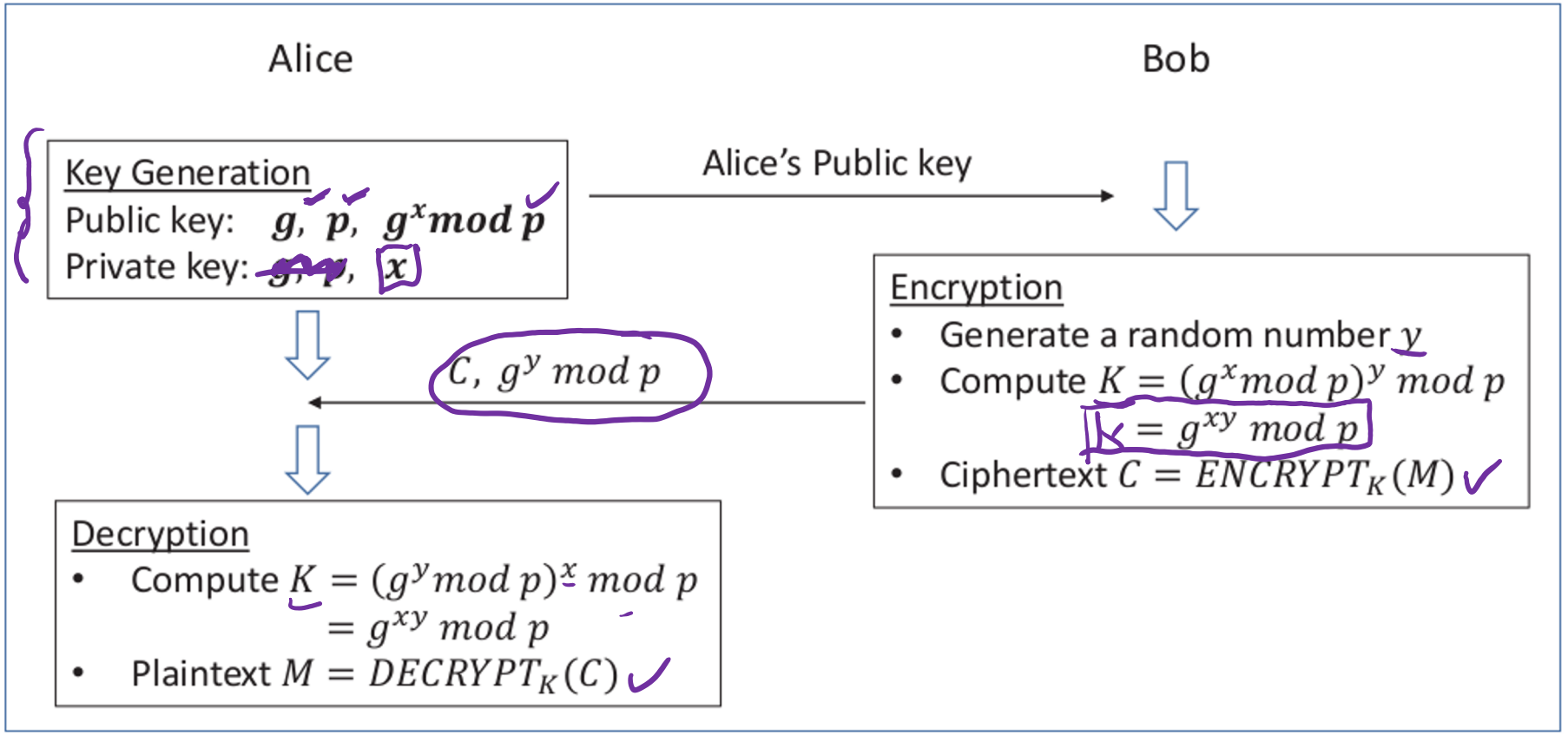


DH public-key encryption

- Originally, DH allows parties to **exchange a secret key**

- What we need
  - **Public key**: known to everyone, used for encryption ✓
  - **Private key**: known to the owner, used for decryption ✓
  - **Algorithm** for encryption and decryption –

DH public-key encryption



RSA math background

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

$$(ab) \bmod n = [(a \bmod n)(b \bmod n)] \bmod n$$

$$a^x \bmod n = (a \bmod n)^x \bmod n$$

Euler's totient function: $\phi(n)$: # of positive integers up to n that are relatively prime to n.

e.g. 2,3 are coprimes
3,4 are coprimes
2,4 are not

a, b are relatively prime (coprimes) iff $\gcd(a, b) = 1$

~~The RSA algorithm~~ background

if p is prime, then $\phi(p) = p - 1$