Cryptography 2

Diffie-Hellman and RSA

Asymmetric Cryptography

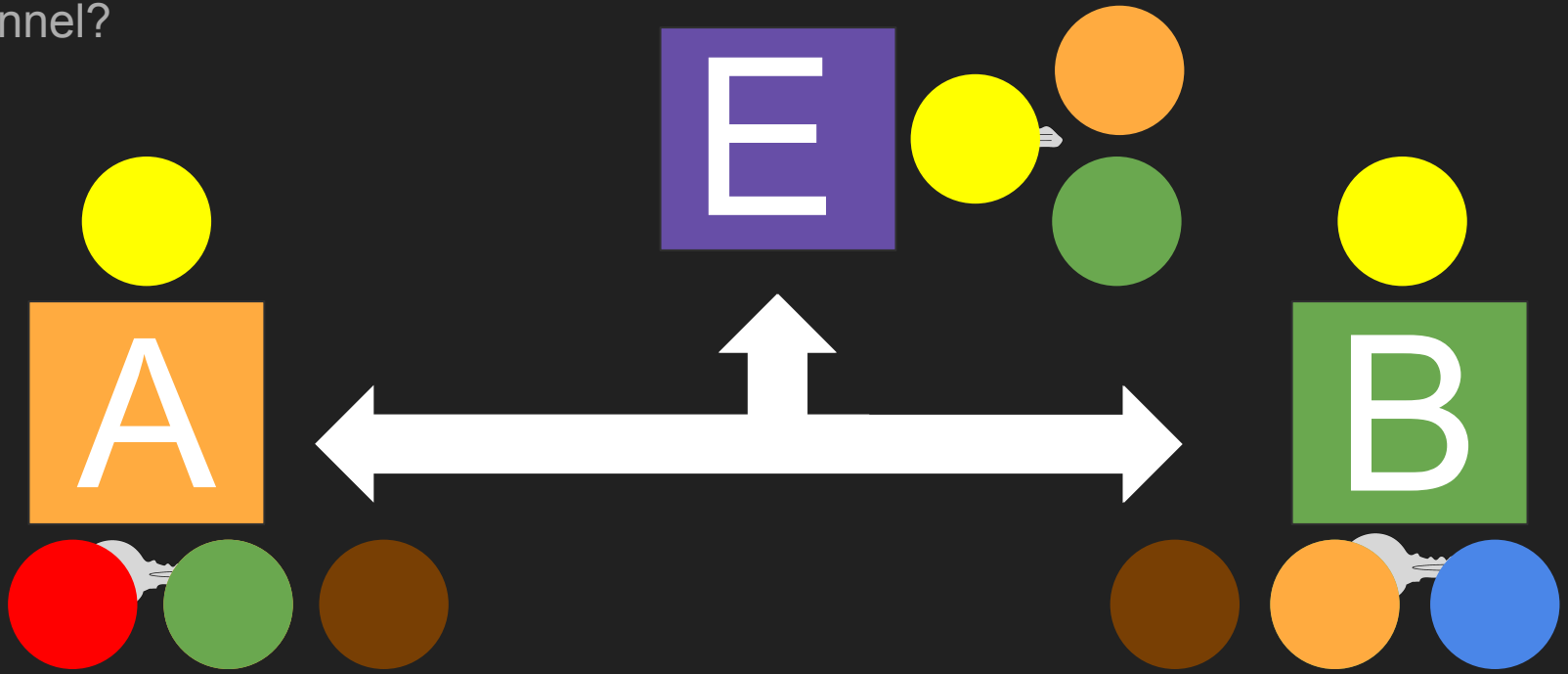
- Parties don’t need to start with a shared secret key
  - Securely sharing a secret key beforehand may not be practical
- Keys are in pairs
  - Public key
    - Can be shared with anyone without compromising
      - Integrity
      - Confidentiality
    - Sharing a public key may have anonymity implications
  - Private key
    - Never shared with anyone
    - Can be stored per device on secure hardware

Uses

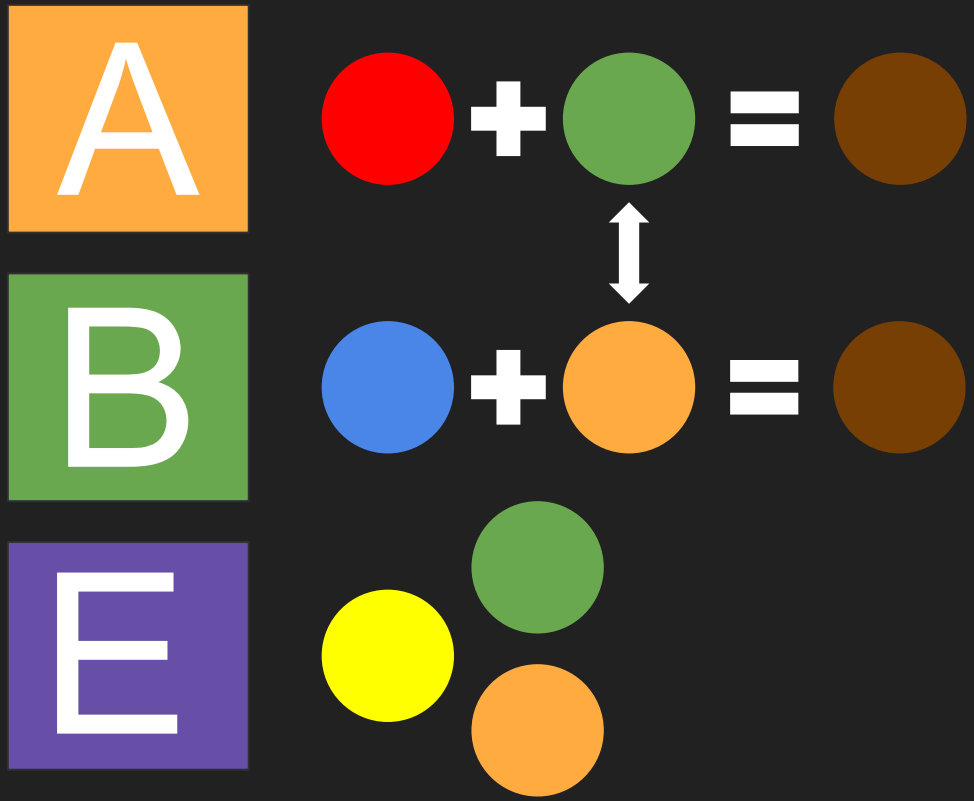
- Key exchange
  - Establish a shared secret key using public parameters
  - Diffie-Hellman is often used for this purpose
- Encryption
  - Public key used to encrypt
  - Private key used to decrypt
  - Hides the true meaning of a message
- Signatures
  - Private key used to sign
  - Public key used to verify
  - Prove the authenticity of a message

Diffie-Hellman Analogy

How do two parties established a shared symmetric key over an untrusted channel?



Diffie-Hellman Analogy



Diffie Hellman

- Relies on one way operations
  - Mixing colors is easy
  - Unmixing colors is hard
- Diffie Hellman can be implemented using modular arithmetic and prime exponentiation
  - Public shared info: large safe prime *p* and a generator *g*
    - *p* is safe if *(p-1)/2* is also a large prime
    - *g* is a generator for *p* if there exists an *n* for every value in *[0, p)* in $g^n \bmod p$
  - Private key: integer *n* in *[0, p)*
  - Public key: $g^n \bmod p = q$

Diffie-Hellman Example

- Alice and Bob agree to use 3 as a generator and 7 as a prime
- Alice picks 2 as a secret number; Bob picks 3
- Alice sends Bob $3^2 \bmod 7 = 2$; Bob sends Alice $3^3 \bmod 7 = 6$
- Alice computes $6^2 \bmod 7 = 1$; Bob computes $2^3 \bmod 7 = 1$
- Eve is left with 3, 7, 2, and 6
- Because the numbers are small, Eve could calculate the shared secret
- In practice, the minimum recommended size of the prime number is 2048 bits

Asymmetric Encryption

RSA Encryption

- Message *m* is encrypted by m^e mod n
- Ciphertext *c* is decrypted by c^d mod n
- e and n form the public key
- d is a secret, but n is also needed during decryption

RSA Signatures

- Prove authenticity of a message
- Sign the message with m^d mod n
- Verify the signature with s^e mod n
- Given the message and the signature, anyone with the public key can verify that the message has not been tampered with

RSA Key Generation

- n, e, and d are not arbitrary
- p and q are large prime numbers
- p * q = n
- p and q are a secret
- Factoring n is really hard
- Given p, q, and e, it is easy to compute d

RSA Key Generation

- $\lambda(n) = \text{lcm}(p - 1, q - 1)$
- e is in $(1, \lambda(n))$
- e must be coprime to $\lambda(n)$
- $d = (e^{-1}) \bmod \lambda(n)$

Limitations of RSA

- Limited message size
- Slower than many symmetric algorithms
- Textbook RSA is vulnerable without padding
- In practice these limitations are avoided by using RSA in addition to other cryptographic primitives
  - A symmetric cipher is typically used for encryption. The key is encrypted with RSA
  - Message hashes are signed instead of the entire message
  - Padding schemes are defined for operations