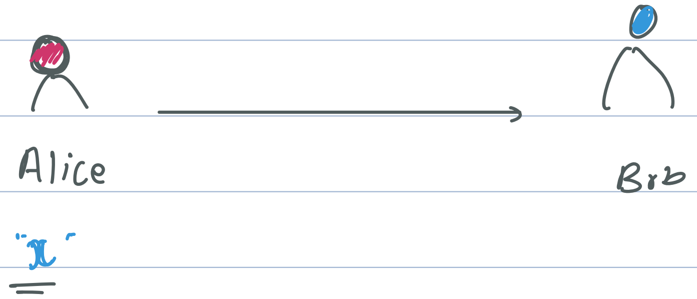

- Alice want's to send message "x" to bob.

Privacy: Privacy ensures only the intended recipient can read the message.
In this case only Bob can read the message x.

1. Symmetric key:- Bob and alice share a symmetric key k

$f(x, k) = c$ $g(c, k) = x$

encryption function decryption function.

2. Assymetric key:

- Alice encrypts x using bob's public key
- Bob decrypts c using his private key.

$f(x, pub.bob) = c$ $f^{-1}(c, private\_bob) = x$.

2. Integrity:- Integrity Ensures that the message has

not been tampered with during transmission.

Using Hash Functions:-

- Alice generates hash of message "x" H(x)
- Alice sends H(x) along with x to bob.

- Bob recieves x and H(x), then he creates H(x) using x and a hashing algo. and matches with H(x).

3. Authenticity :-

1. Ensures that message x was indeed sent by Alice
2. Message is not forged by third party.

- Alice digitally signs H(x) using her private key.

- Bobs, using Alice's public key, gets H(x)
  this proves that message was indeed sent by Alice

Combining the Process :-

Alice

Privacy Encryption Function

x

↓ encrypts using Bob's public key

C

↓

Sends this to Bob

x

↓ Hash

H(x)

Hash function for integrity

↓ using he· private key she signs H(x)

Digital Signature Algo.

S

↓

for authenticity

sends this to Bob

Bob

(Bob recieves C and S

message from Alice

C ⟶ x

using his private key Bob decrypts C

confirms privacy but not integrity or authenticity

Authentication

S ⟶ H(x)

using Alice's public key.

using alice's public key anyone can get H(x)

Bob compares the Hash

This is for integrity

x ⟶ H(x)

using hash

function bob

hashes λ