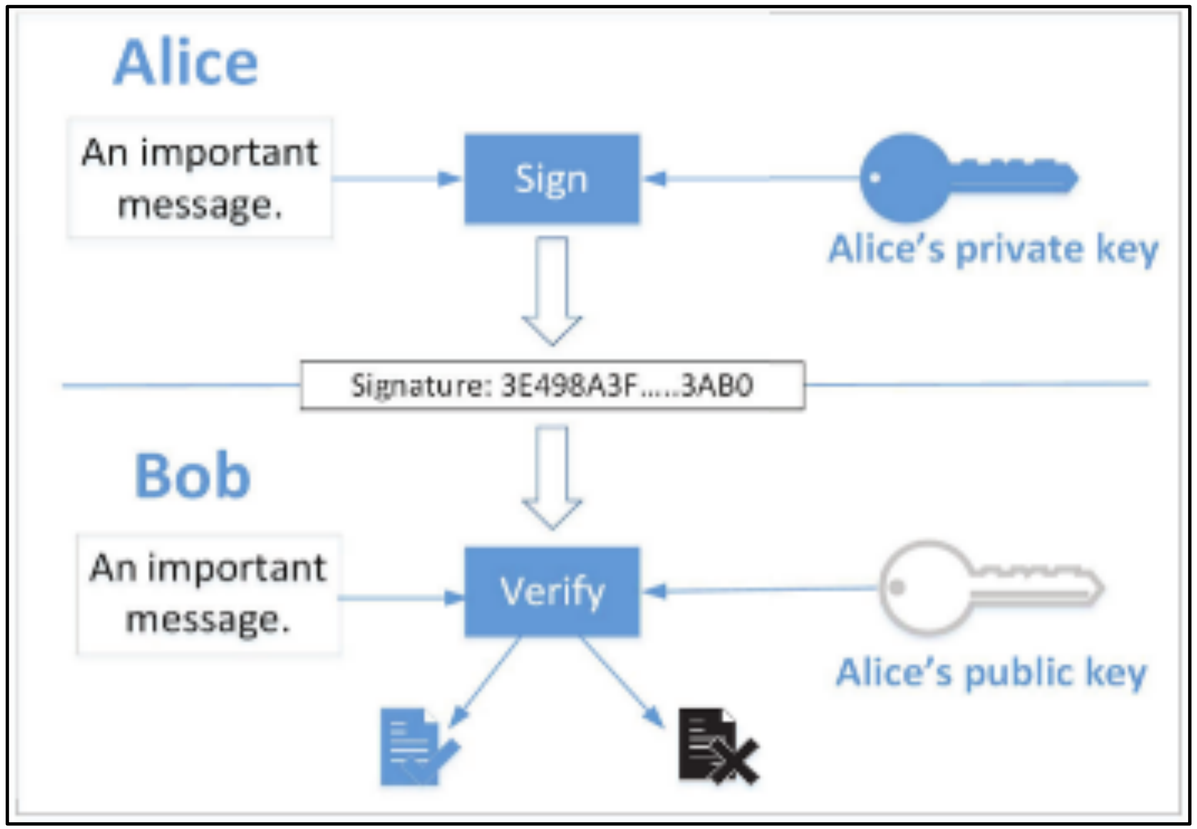
Digital Signatures

This Video Covers:

- Digital signatures: *what are they?* and *how do they work?*
- Examples with OpenSSL
- Experiment: attacks on digital signatures

Digital Signatures

- **Goal:** provide *proof of authenticity* by signing digital documents
  - Diffie-Hellman authors proposed the idea, but no concrete solution
  - RSA authors developed the first digital signature algorithm



Digital Signature using RSA

- Apply private-key operation on m using private key, and get a number s
  (anyone can get m back from s using the public key)

- To sign a message m:
  - Digital signature = $m^d \bmod n$

- In practice, a message may be long resulting in a long signature
  and more computing time

- Instead, we generate a cryptographic hash value from the original message,
  and only sign the hash

Digital Signature using RSA *(cont.)*

To generate a hash of the message:

```
# Generate a sha256 hash of the secret message
$ openssl sha256 -binary msg.txt > msg.sha256

$ xxd msg.sha256
00000000: 8272 61ce 5ddc 974b 1b36 75a3 ed37 48cd  .ra.]..K.6u..7H.
00000010: 83cd de93 85f0 6aab bd94 f50c db5a b460  ……j……Z.`
```

Digital Signature using RSA *(cont.)*

To generate and verify the signature:

```
# Sign the hash
$ openssl rsautl -sign -inkey private.pem -in msg.sha256 -out msg.sig

# Verify the signature
$ openssl rsautl -verify -inkey public.pem -in msg.sig -pubin -raw | xxd
00000000: 0001 ffff ffff ffff ffff ffff ffff ffff  ................
00000010: ffff ffff ffff ffff ffff ffff ffff ffff  ................
00000020: ffff ffff ffff ffff ffff ffff ffff ffff  ................
00000030: ffff ffff ffff ffff ffff ffff ffff ffff  ................
00000040: ffff ffff ffff ffff ffff ffff ffff ffff  ................
00000050: ffff ffff ffff ffff ffff ffff ffff ff00  ................
00000060: 8272 61ce 5ddc 974b 1b36 75a3 ed37 48cd  .ra.]..K.6u..7H.
00000070: 83cd de93 85f0 6aab bd94 f50c db5a b460  ......j......Z.`
```

Attack Experiment on Digital Signatures

- Attackers cannot generate a valid signature from a modified message because they do not know the private key

- If an attacker modifies the message, the hash will change, and therefore the signature of the hash will change, so the signature verification should fail

Experiment:

Modify 1 bit of the signature file `msg.sig`
and verify the signature

Attack Experiment on Digital Signatures

After applying the RSA public key on the signature,
we get a block of data that is significantly different

```
$ openssl rsautl -verify -inkey public.pem -in msg.sig -pubin -raw | xxd
00000000: 07a4 8d1c cfb8 b36c 17af e821 a9ea 8c80  .......l...!....
00000010: c654 74b0 afb1 c1d8 616c 9dca 5138 3b9d  .Tt.....al..Q8;.
00000020: 8111 234e d20f 033f 07f2 7f7c a88e 4fb1  ..#N...?...|..O.
00000030: 14e0 8132 6b6e ae1e 2a4c be54 ff61 f2e6  ...2kn..*L.T.a..
00000040: 965e 492c 428a 2cd3 8c07 7764 480d 2697  .^I,B.,...wdH.&.
00000050: db36 f2a4 7916 27aa 8a07 17c4 d94a 1f06  .6..y.'......J..
00000060: 2632 cf4b fb2c e98f fb68 cbe1 b084 3bb1  &2.K.,...h....;.
00000070: bb98 651c 0469 14f5 2f92 0e91 93d7 2d09  ..e..i../.....-.
```