Digital Signatures

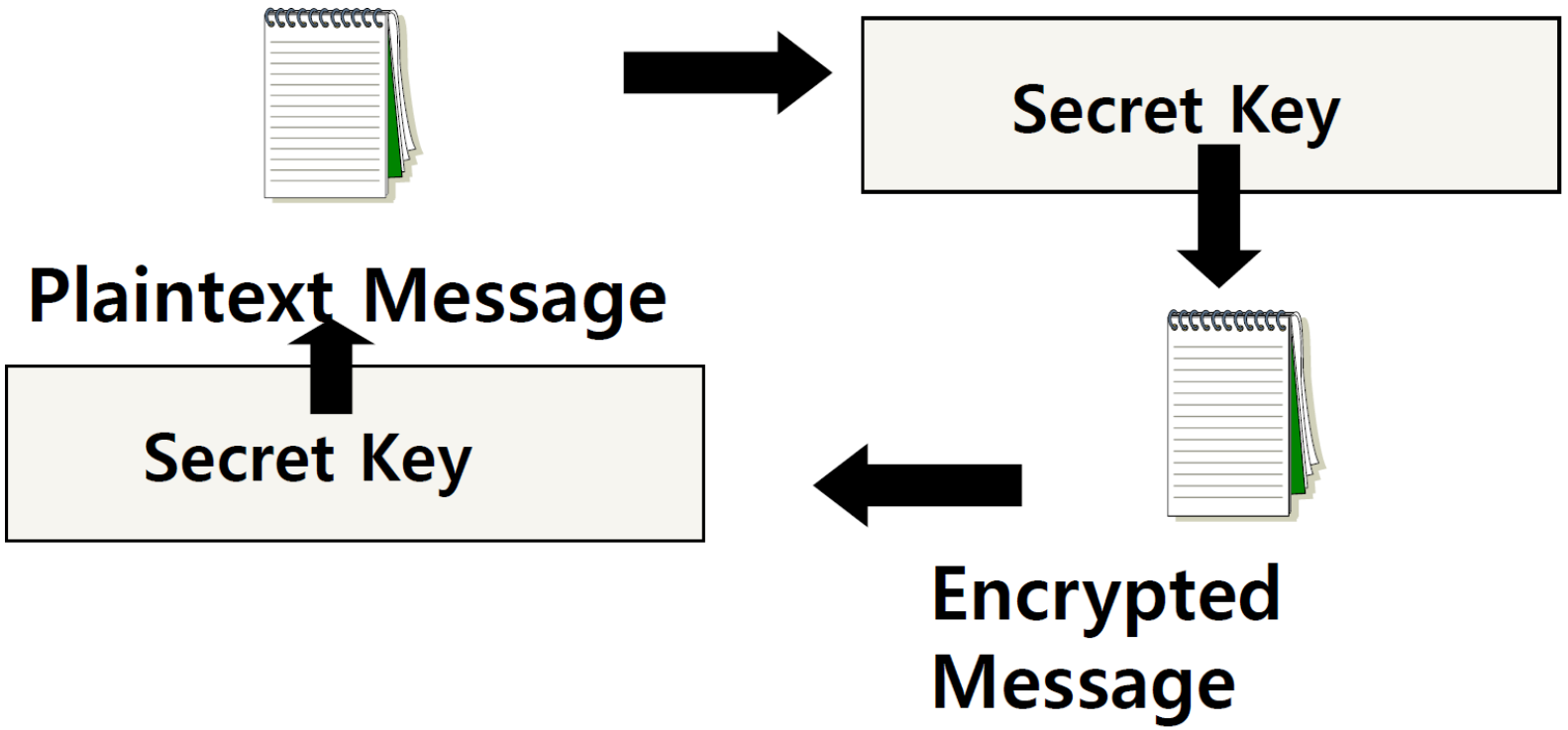
Homework 6

Communication Over the Internet

- What type of guarantees do we want?
  - **Confidentiality**
    - Message secrecy
  - **Data integrity**
    - Message consistency
  - **Authentication**
    - Identity confirmation
  - **Authorization**
    - Specifying access rights to resources

Secret Key (symmetric) Cryptography

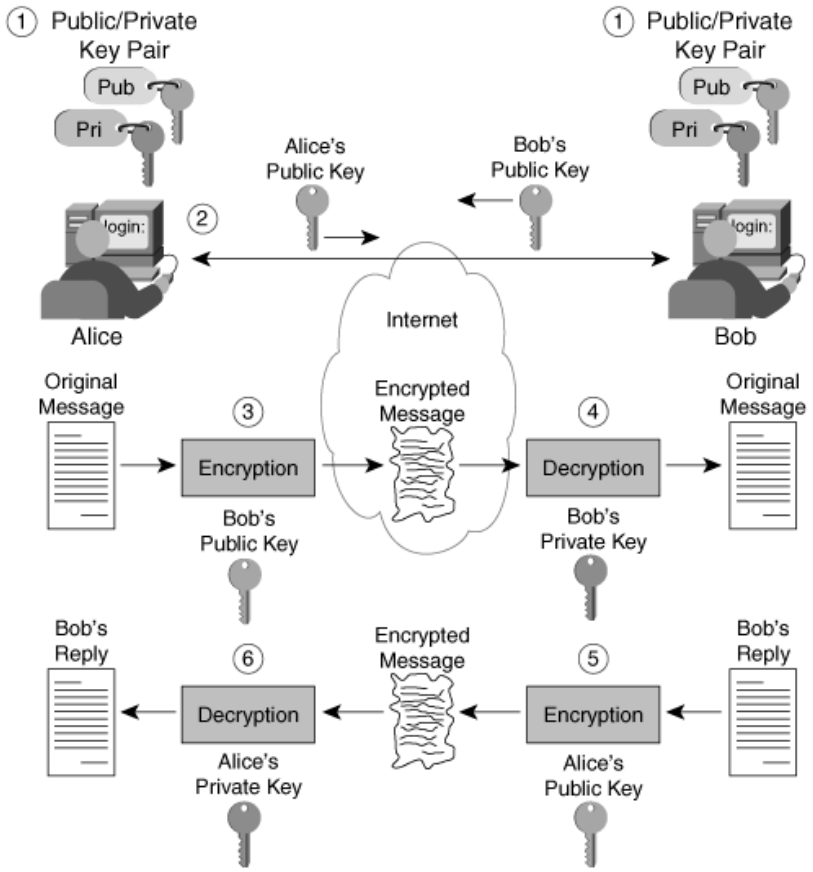
- A single key is used to both encrypt and decrypt a message



Public Key (asymmetric) Cryptography

- Two keys are used: a public and a private key.
  - If a message is encrypted with one key, it has to be decrypted with the other.

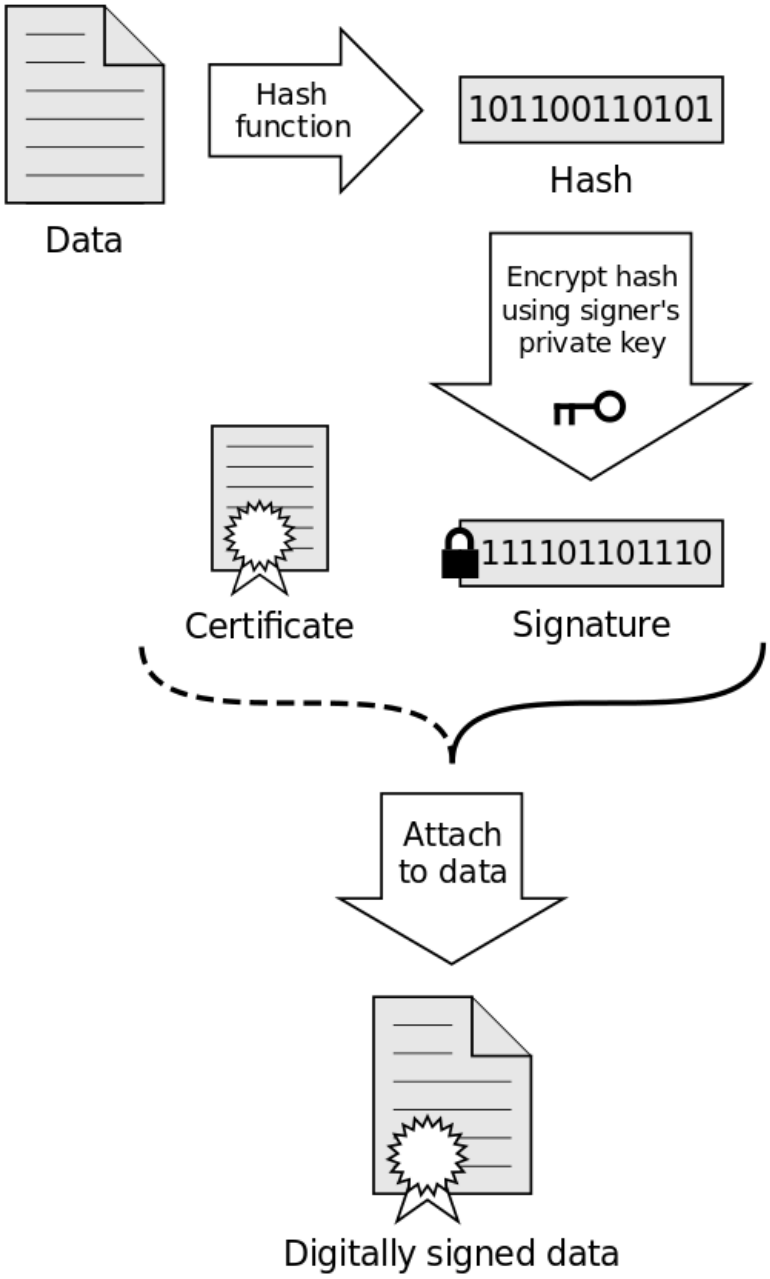
Key used for encryption depends on goal.



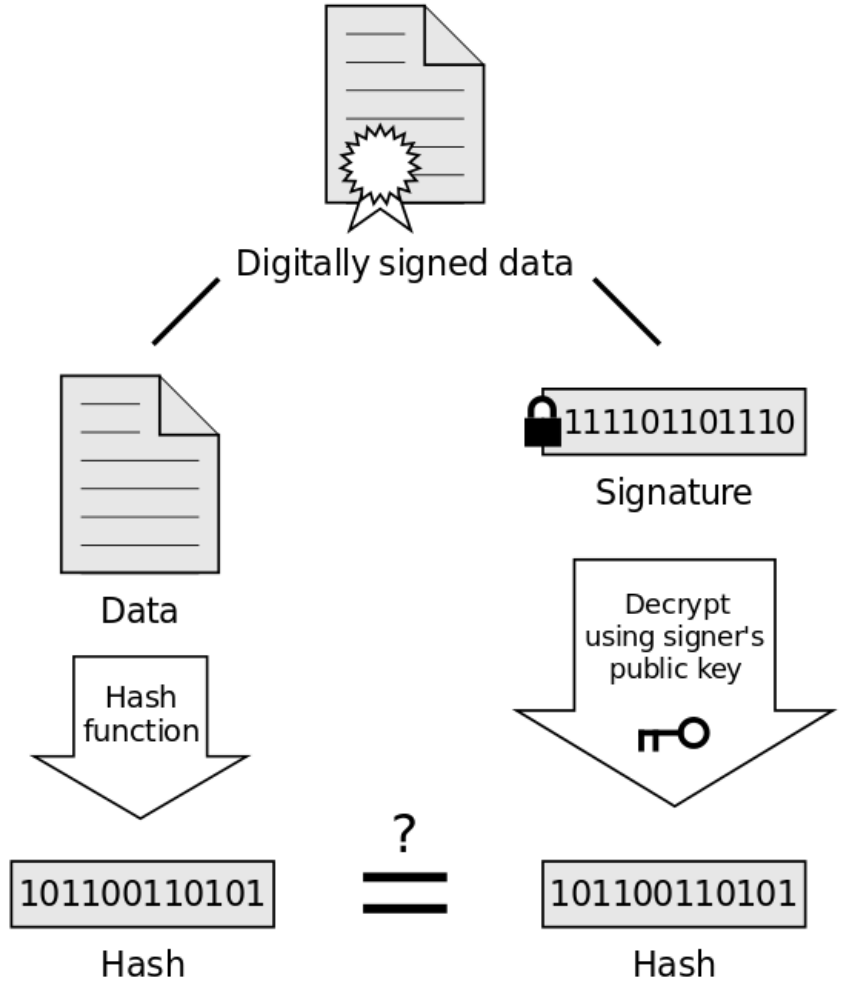
Digital Signature

- An electronic stamp or seal, which is appended to a document
- Ensures **data integrity**
  - document was not changed during transmission

Signing



Verification



If the hashes are equal, the signature is valid.

Steps for Generating a Digital Signature

SENDER:

1) Generate a *Message Digest*
   - The message digest is generated using a set of hashing algorithms
   - A message digest is a 'summary' of the message we are going to transmit
   - Even the slightest change in the message produces a different digest
2) Create a Digital Signature
   - The message digest is encrypted using the sender's *private* key. The resulting encrypted message digest is the *digital signature*
3) Attach digital signature to message and send to receiver

Steps for Generating a Digital Signature

RECEIVER:

1) Recover the *Message Digest*
   - Decrypt the digital signature using the sender's public key to obtain the message digest generated by the sender
2) Generate the Message Digest
   - Use the same message digest algorithm used by the sender to generate a message digest of the received message
3) Compare digests (the one sent by the sender as a digital signature, and the one generated by the receiver)
   - If they are not *exactly the same* => the message has been tampered with by a third party
   - We can be sure that the digital signature was sent by the sender (and not by a malicious user) because *only* the sender's public key can decrypt the digital signature and that public key is proven to be the sender's through the certificate. If decrypting using the public key renders a faulty message digest, this means that either the message or the message digest are not exactly what the sender sent.

Detached Signature

- Digital signatures can either be *attached* to the message or *detached*
  - A detached signature is stored and transmitted separately from the message it signs
  - Commonly used to validate software distributed in compressed tar files
  - You can't sign such a file internally without altering its contents, so the signature is created in a separate file

Homework 6

- Answer 2 questions in the file **`hw.txt`**
- Generate a key pair with the GNU Privacy Guard's commands
  - `$ gpg --gen-key` (choose default options)
- Export public key, in ASCII format, into **`hw-pubkey.asc`**
  - `$ gpg --armor --output hw-pubkey.asc --export 'Your Name'`
- Make a tarball of the above files + **`log.txt`** and zip it with gzip to produce `hw.tar.gz`
  - `$ tar -cf hw.tar <files>`
  - `$ gzip hw.tar` -> creates hw.tar.gz
- Use the private key you created to make a detached clear signature `hw.tar.gz.sig` for `hw.tar.gz`
  - `$ gpg --armor --output hw.tar.gz.sig --detach-sign hw.tar.gz`
- Use given commands to verify signature and file formatting