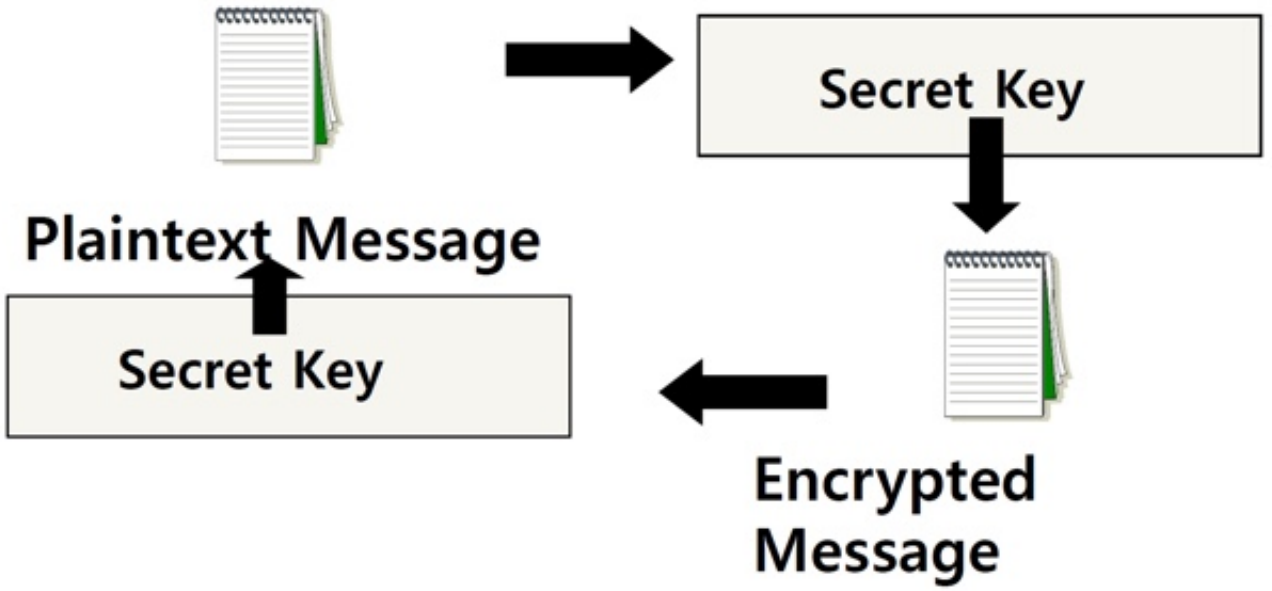
Week 6: Digital Signatures

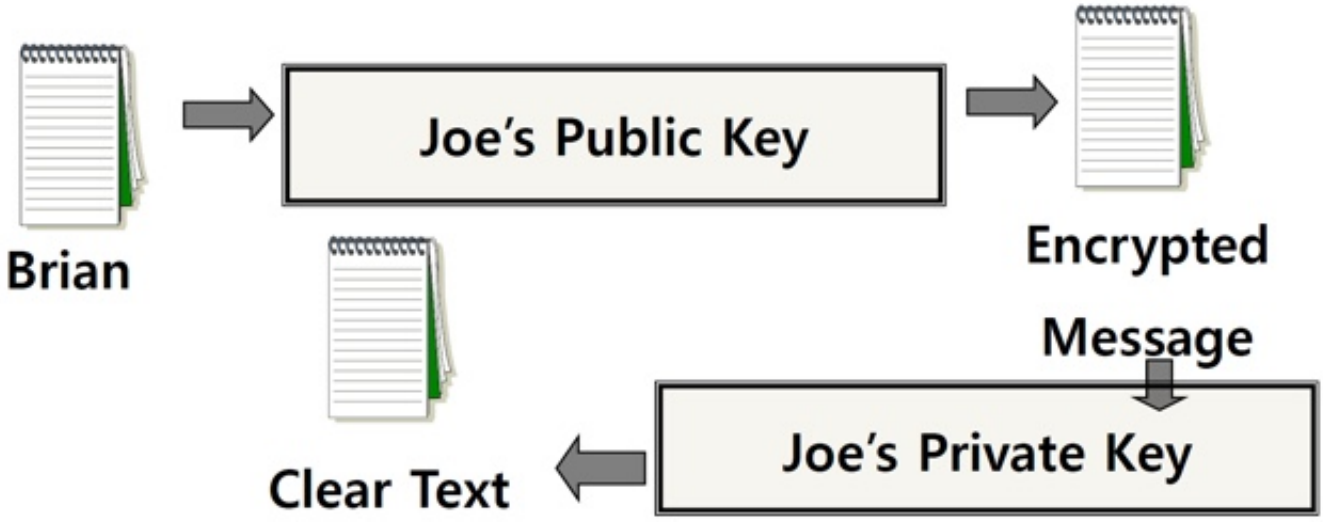
Secret Key (symmetric) Cryptography

- 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



Digital Signature

- an electronic stamp or seal
    - almost esactly like a written signature, except more guarantees
- is appended to a document
    - or sent separately (detached signature)
- ensures data integrity
    - document was not changed during transmission

Generating a Digital Signature

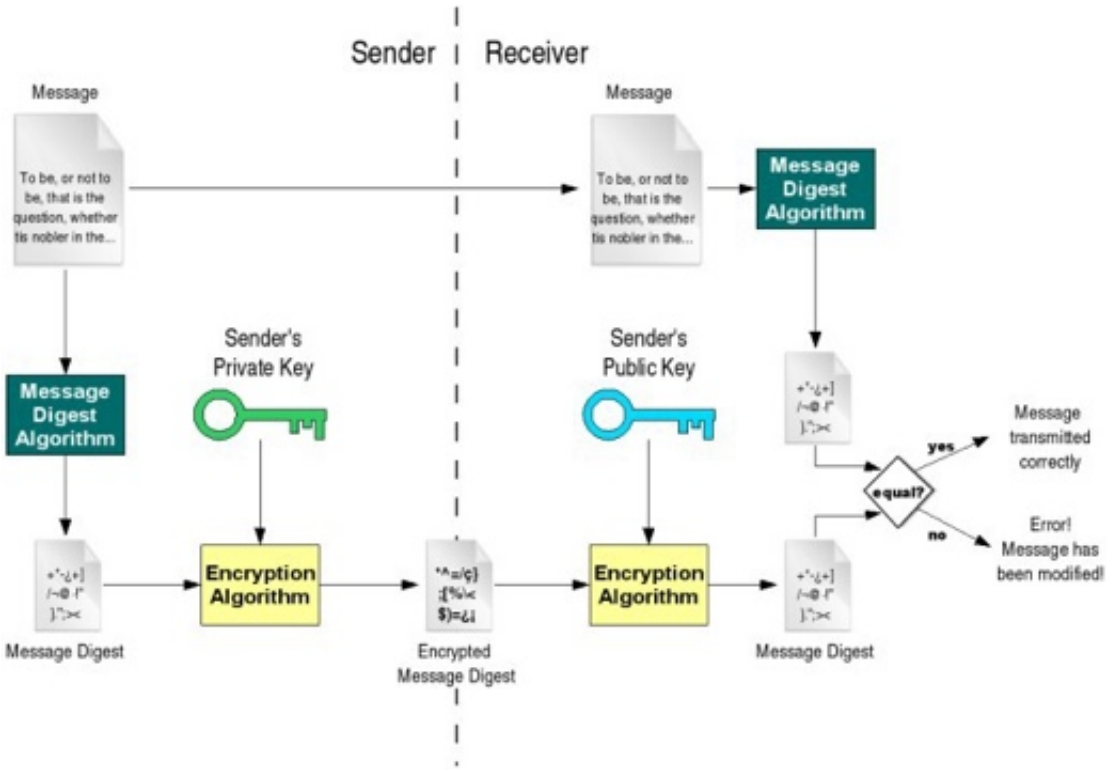
SENDER

1. Generate a message digest
    a. generated using a set of hashing algorithms
    b. a message digest is a ‘summary’ of the message we are going to transmit
    c. even the slightest change in the message produces a different digest
2. Create a digital signature
    a. message digest is encrypted is encrypted using the sender’s *private* key
    b. resulting encrypted message is the **digital signature**
3. Attach digital signature to message and send to receiver

RECEIVER

1. Recover the message digest
    a. decrypt the digital signature using the sender’s public key to obtain the message digest generated by the sender
2. Generate the message digest
    a. 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)
    a. if they are not *exactly* the same ➤ message has been tampered with by a third party
    b. we can be sure that the digital signature was sent by the sender 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

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
- 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
  - These can be found at the end of the assignment spec