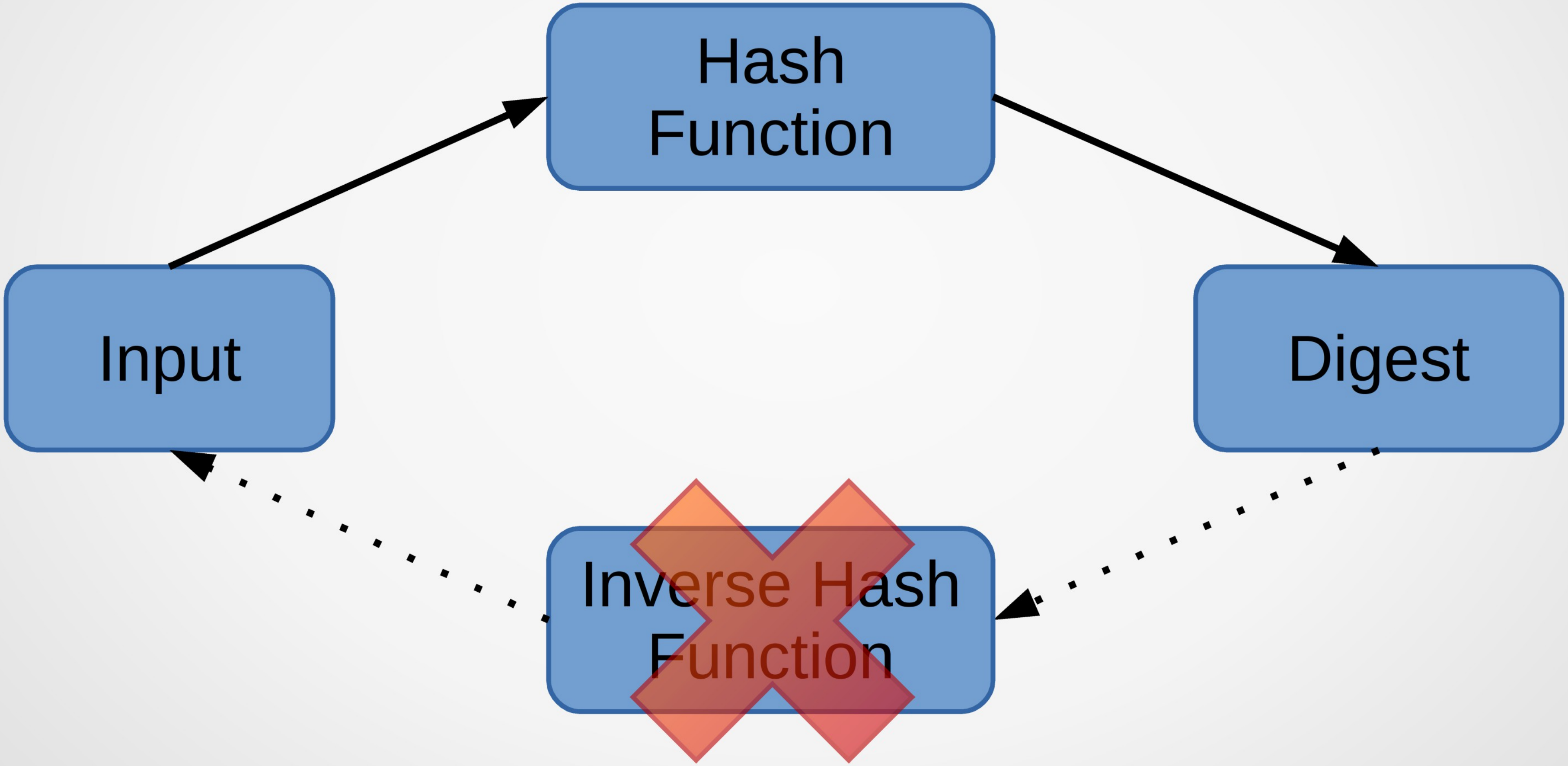
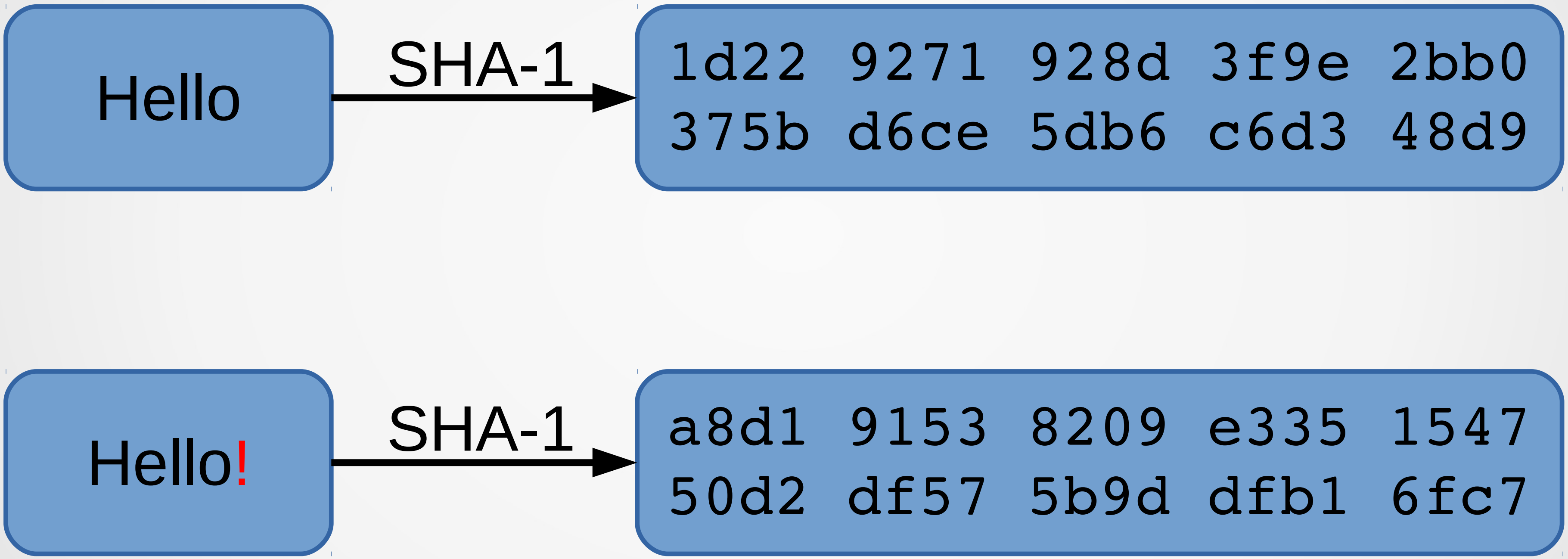
Cryptography

Hash Function



Example Hash Values



Hello

SHA-1

1d22 9271 928d 3f9e 2bb0
375b d6ce 5db6 c6d3 48d9

Hello!

SHA-1

a8d1 9153 8209 e335 1547
50d2 df57 5b9d dfb1 6fc7

Properties of Hash Function

- Easy to compute the hash value
- Infeasible to generate original message from hash
- Infeasible to modify message without changing hash
- Infeasible to find two messages with same hash

Storing Passwords

- Data falls into hands of adversaries
- Passwords should not be stored in plain text
- Passwords hashes must be stored instead of plain passwords

Rainbow Table Attacks

- Tables of precomputed hashes can be used
- Password for a given hash can then be easily obtained
- Random salts are added to the passwords to prevent this attack
- Random salt is stored along with password hash
- Repeating the hashing algorithm 1000s of times makes computing rainbow tables difficult

Brute Force Attacks

- Adversary simply tries all possible passwords
- Online attack means trying passwords in a weak system
- Offline attack involves obtaining password hashes from the system and computing hashes of various passwords
- Passwords need to be strong to stop such attacks

Password Strength

- 1 bit = 2 possibilities
- 1 byte = 256 possibilities
- 1 alphabet = 26 possibilities
- 8 alphabets = 26 ** 8 ≈ 208 million possibilities
- 1 dictionary word ≈ 100,000 possibilities

Password Strength

- 1 alphabet = 26 possibilities
- 8 alphabets = 26 ** 8 ≈ 208 million
- 1 upper/lower case alphabet = 52
- 8 upper/lower case alphabets = 52 ** 8 ≈ 53 trillion
- 1 alpha or digit = 62
- 8 alpha or digits = 62 ** 8 ≈ 218 trillion
- 1 alpha, digit or special = 72 (say)
- 8 alpha, digit or specials = 72 ** 8 ≈ 722 trillion
- Password with proper characters is millions of times harder to crack

Password Strength

- 1 dictionary word ≈ 100,000 possibilities
- Variations such as capitalizations and digit replacement don't add much
- Digit and symbols at the start/end contribute very little
- We end up with less than 1 billion possibilities
- On the web this is 11 days with thousand possibilities per second
- Offline, when some is comparing hashes, it is just a few seconds

Random Passwords

- 1 alpha, digit or symbol = 72 possibilities
- 8 alpha, digit or symbols ≈ 722 trillion possibilities
- 16 alpha, digit or symbols ≈ 521 thousand trillion trillion possibilities
- Difficult to remember

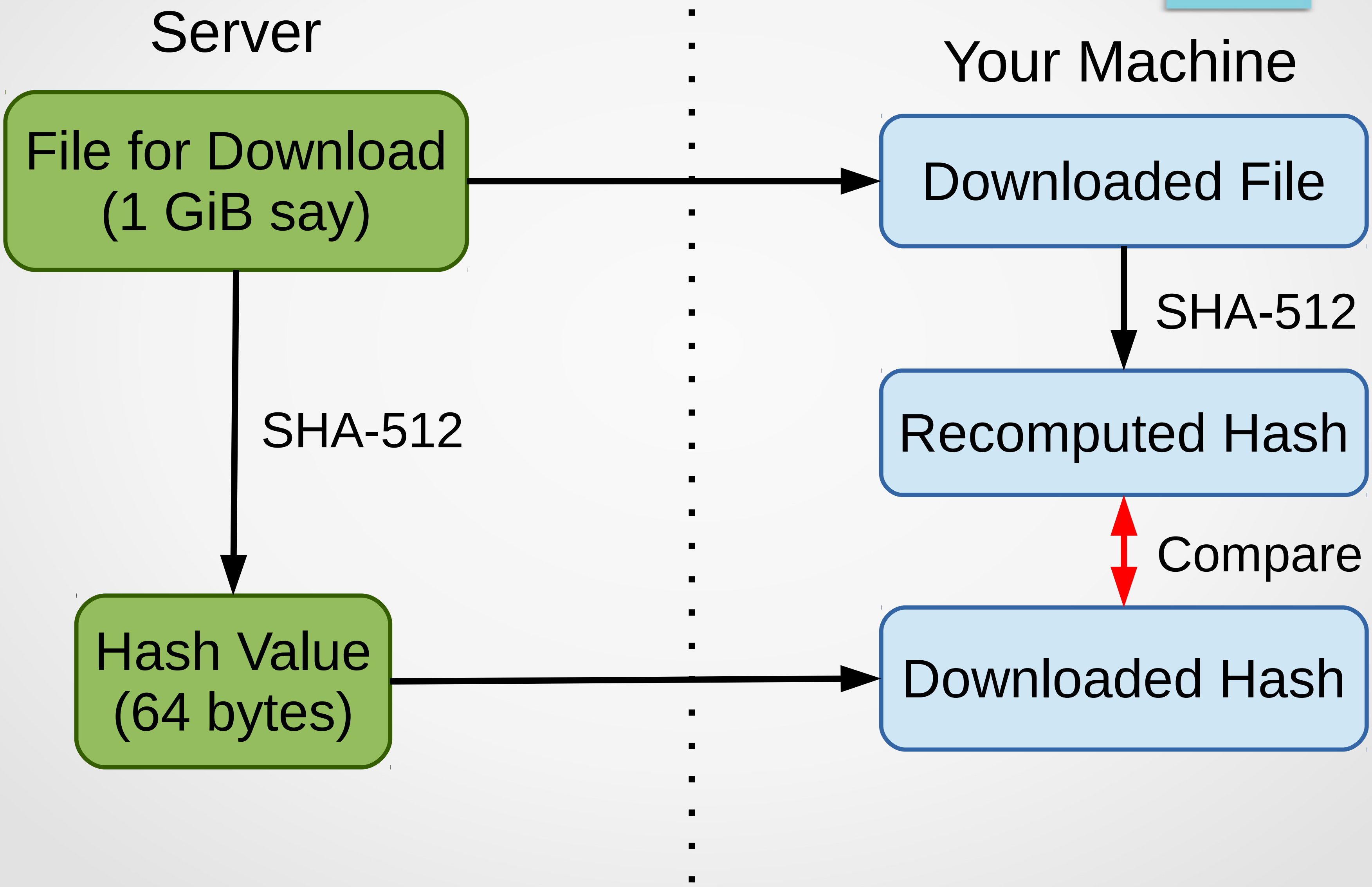
Passphrase Strength

- 1 dictionary word ≈ 100,000 possibilities
- 4 dictionary words ≈ 100 million trillion possibilities
- Much easier to remember than single word weirdly twisted

Password Management

- We have dozens/hundreds of accounts and passwords
- Generate large random passwords
- Use software/hardware password manager
- Good examples:
  - A simple encrypted file (if properly handled)
  - Firefox password manager (with master password set)
- Bad examples:
  - Online password storage services

Checking Data Integrity



Checking Data Integrity

```
kirk@ent:~$ echo Hello > hello.txt
kirk@ent:~$ sha256sum hello.txt
66a045b4521...2c1bb35f18  hello.txt
kirk@ent:~$ sha256sum hello.txt > SHA256SUMS

kirk@ent:~$ sha256sum -c SHA256SUMS
hello.txt: OK
```

Other Uses of Hashes

- File synchronization
- Indexes for efficient data retrieval
- De-duplication of data stored or backed up

Popular Hashing Algorithms

- SHA – 3 (recently selected)
- SHA – 2 (512) (recommended)
- SHA – 2 (256)
- SHA – 1 (known attacks)
- MD5 (known attacks, collisions found)

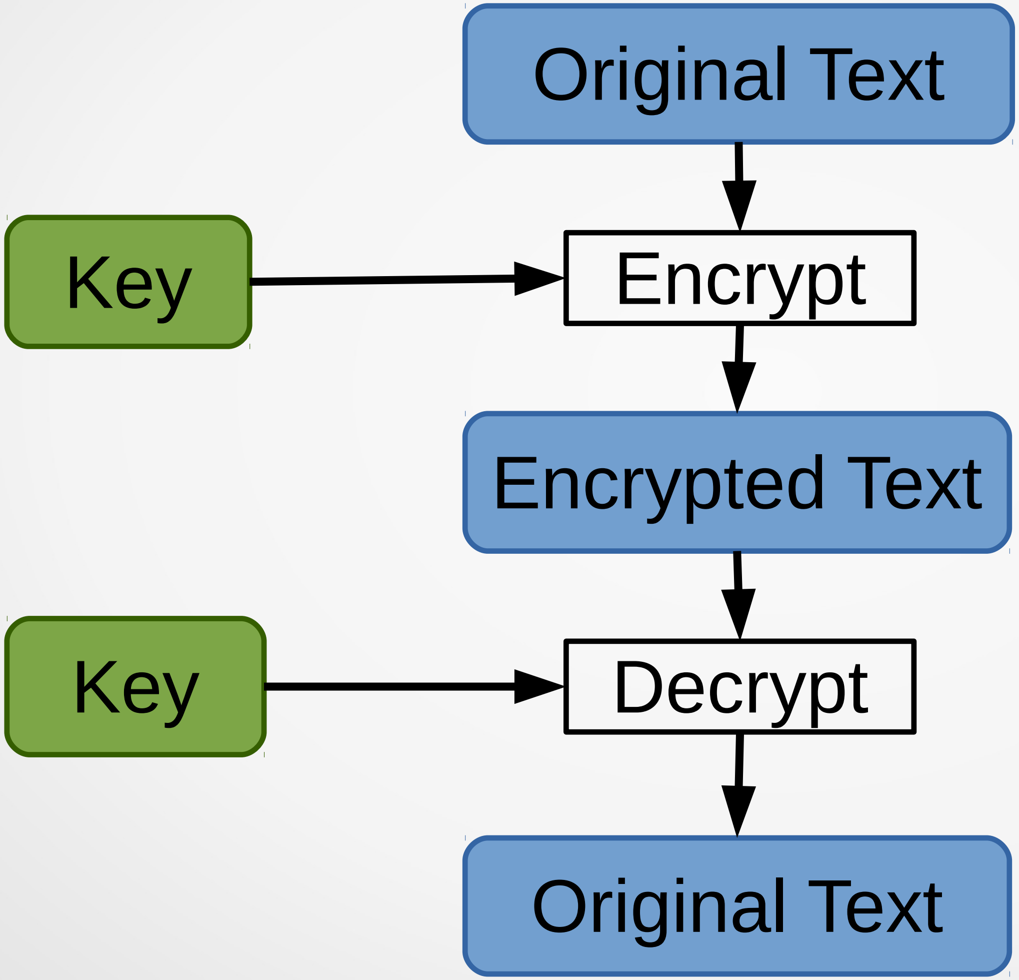
Encryption

- Changing a message into unreadable apparent nonsense
- So that only authorized parties can read

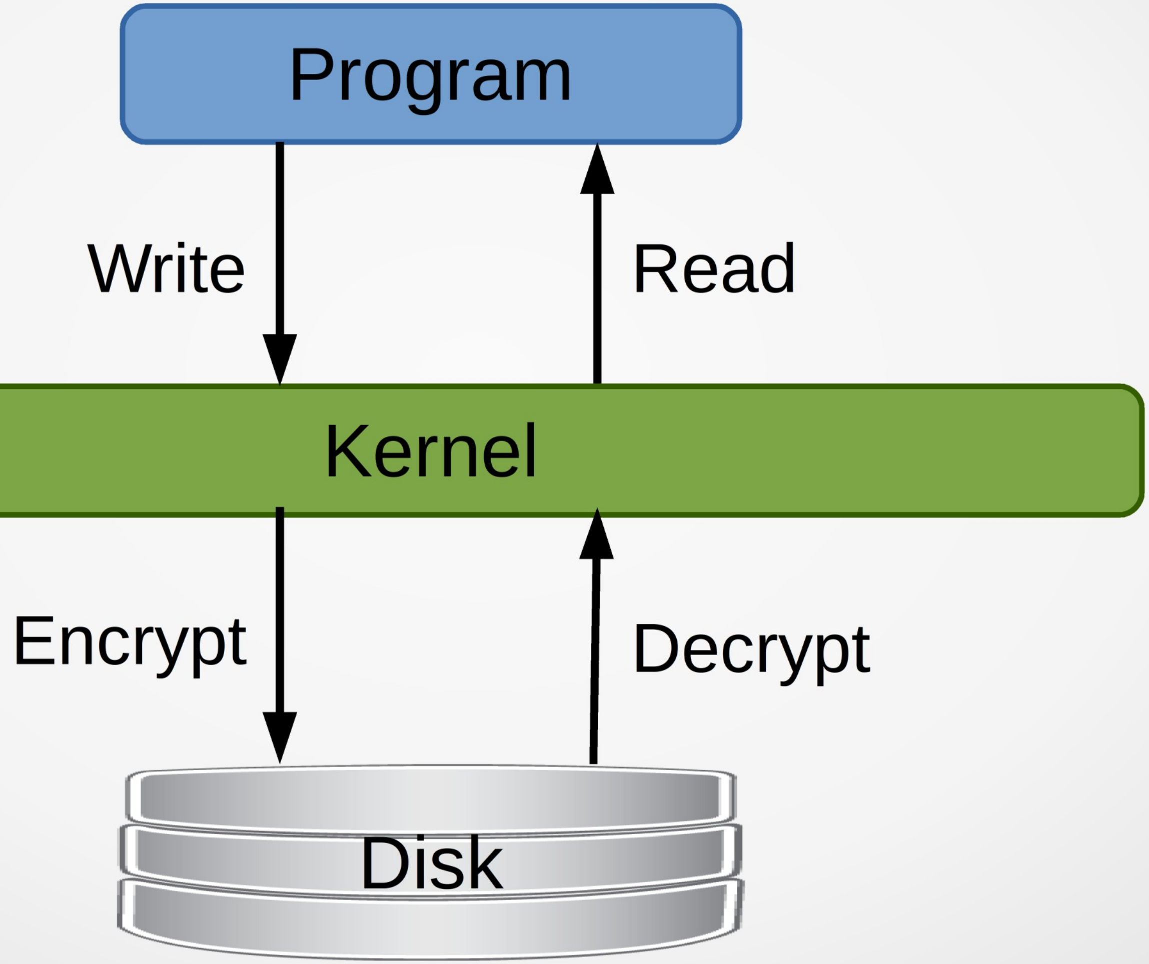
Decryption

- Extracting the original message from encrypted text

Symmetric Key Encryption



Full Disk Encryption


Program

Write

Read

Kernel

Encrypt

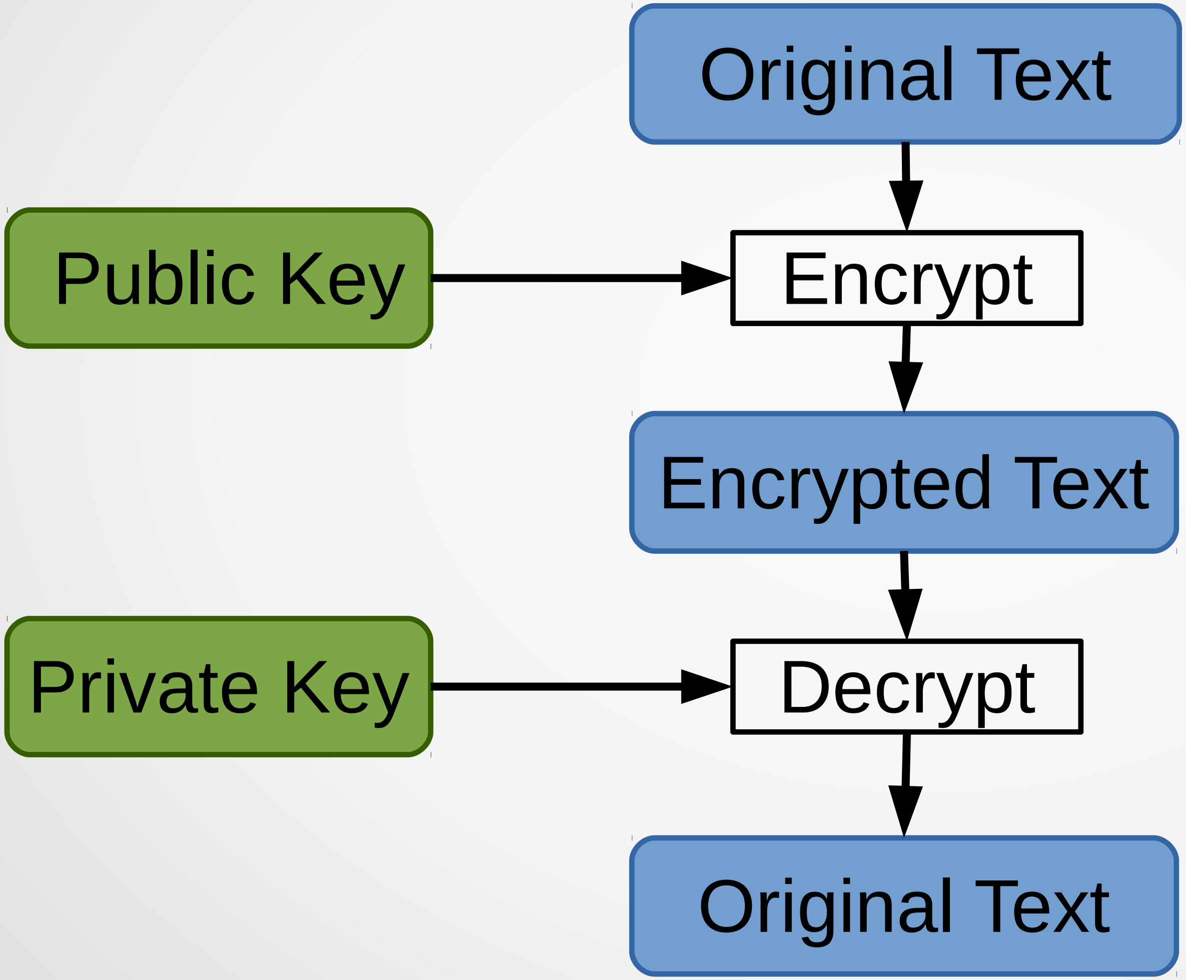
Decrypt

Disk

Full Disk Encryption

- When formatting the disk, choose to encrypt
- Can use a password or a key
- Need to provide decryption key/password during usage
- Negligible CPU overhead for encryption/decryption
- Makes erasing disks safe and easy

Public Key Encryption



Original Text

Public Key

Encrypt

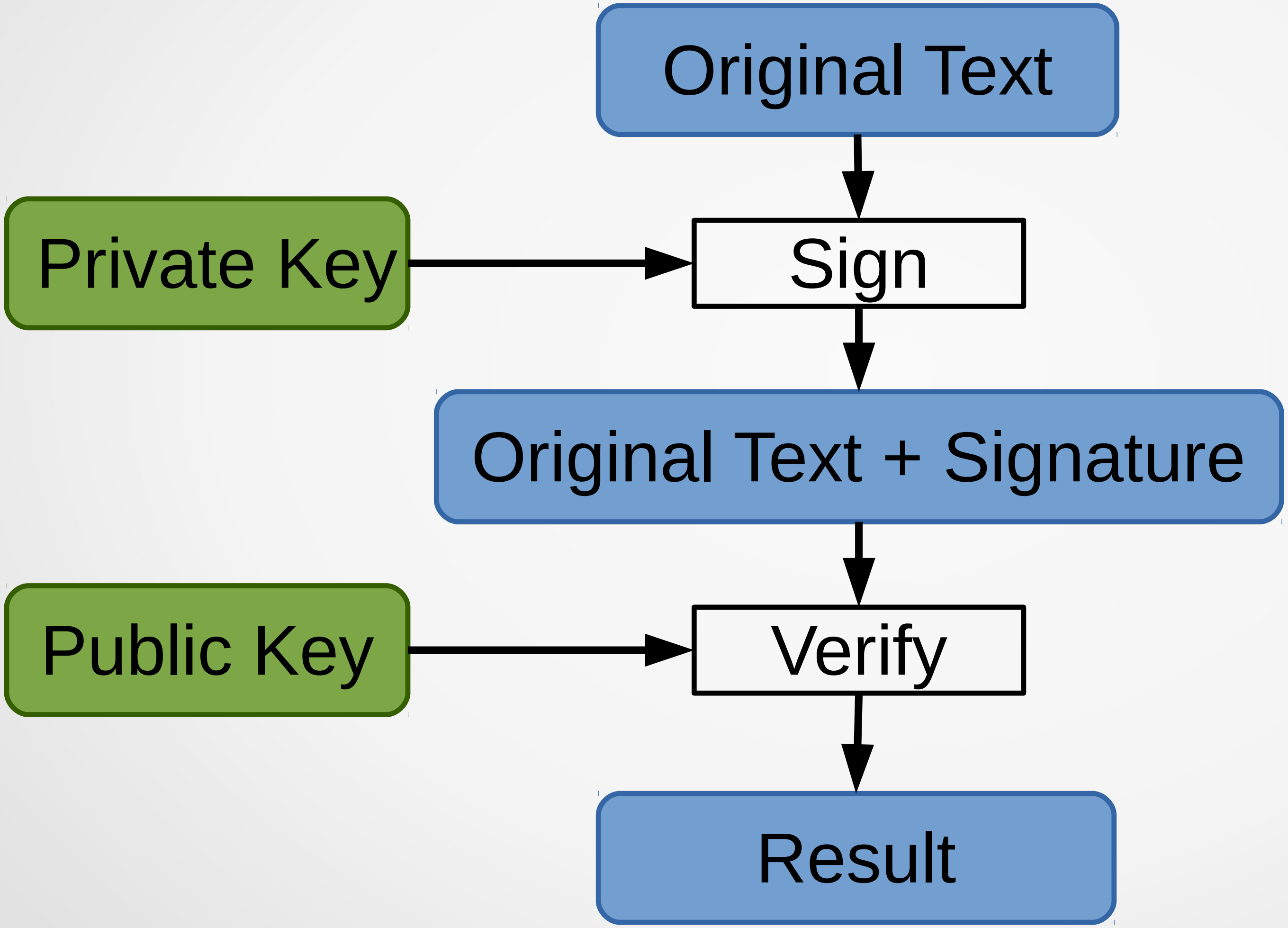
Encrypted Text

Private Key

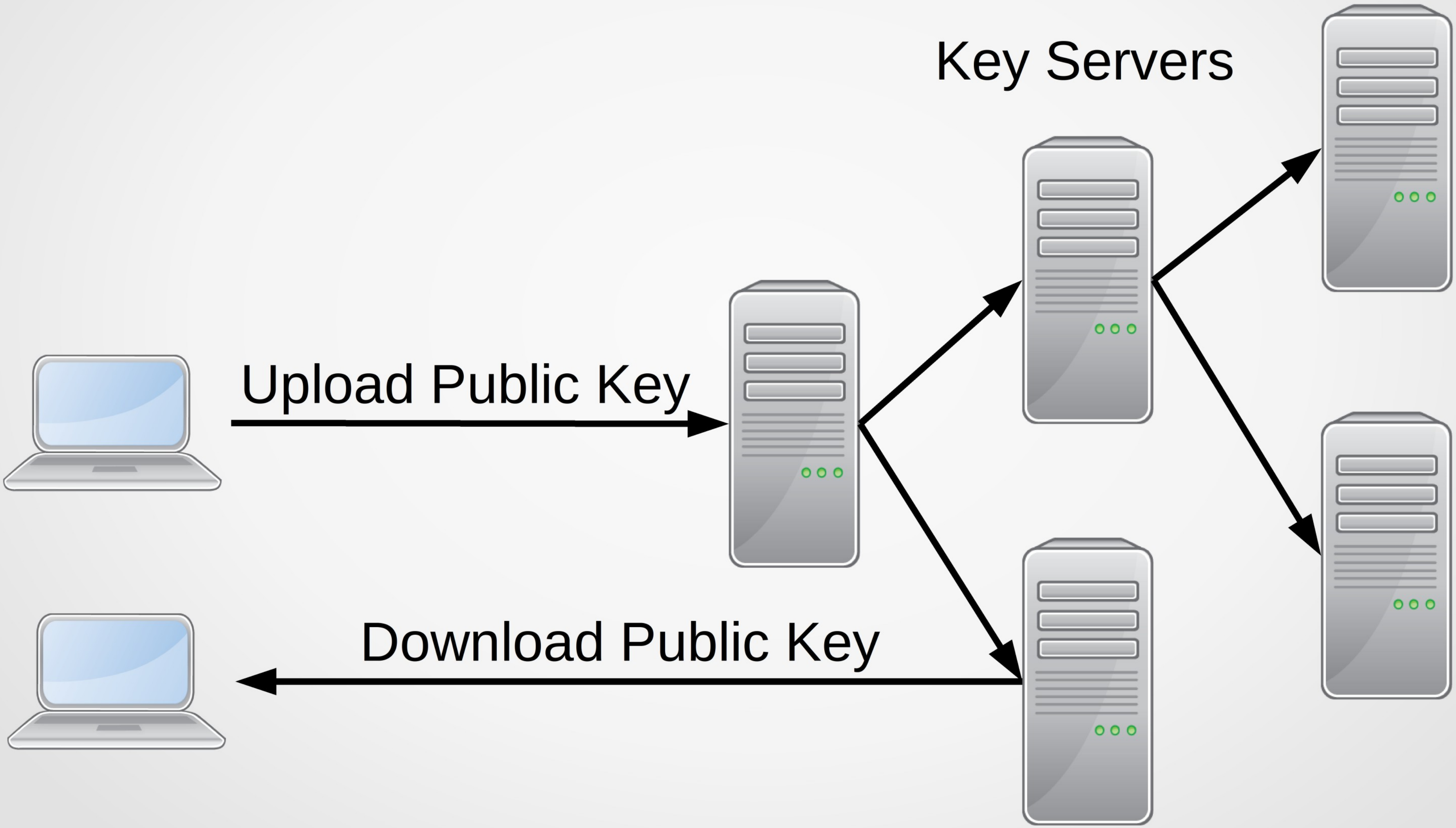
Decrypt

Original Text

Public Key Signing



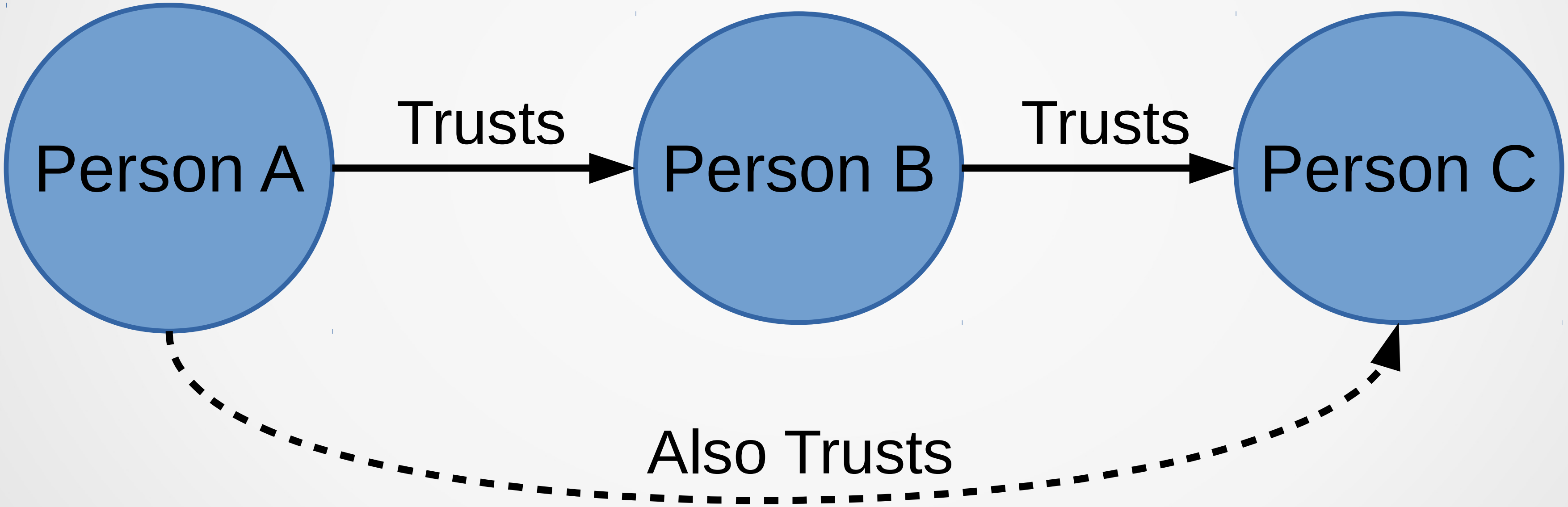
Key Servers



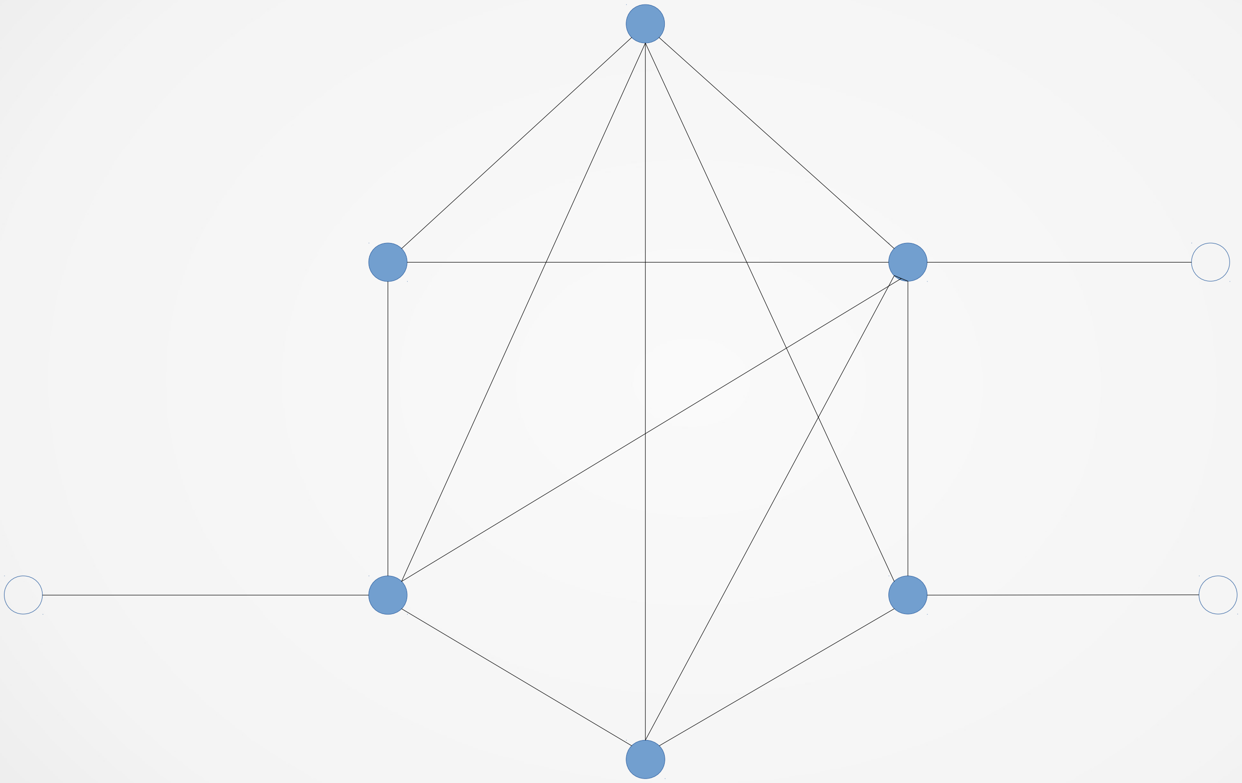
Key Signing

- Public keys have to be verified properly
  - Verify identity of a person (using identity documents)
  - Receive fingerprint of their key
  - Download their public key
  - Sign their key with your private key
  - Now you trust them
- Distribute your trust
  - Upload your signature to key servers
  - Other's now know you trust them

Web of Trust



Web of Trust



GNU Privacy Guard (GPG)

- Generate public/private key pair
- Upload public keys/signatures to server
- Download others' keys
- Sign keys
- Sign/verify message
- Encrypt/decrypt messages

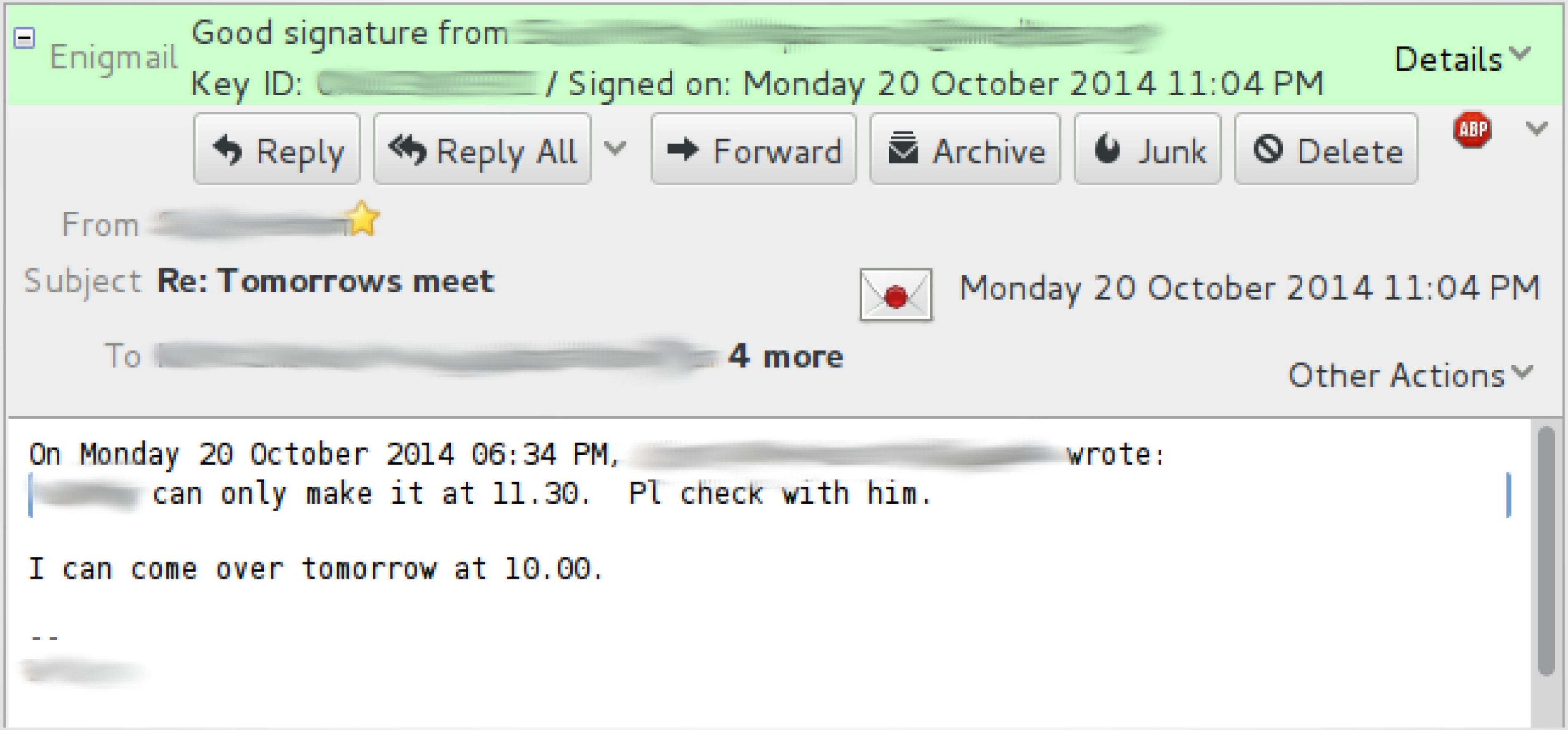
Thunderbird & Enigmail

- Thunderbird is a desktop GUI email client
- Enigmail is an GPG addon for Thunderbird
- OpenPGP compliant, send mails to any such client
- Uses GPG internally
- Sign/verify email messages
- Encrypt/decrypt email messages
- Manage GPG keys using a GUI

Email Signing

```
-----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA1

My Dear Watson, How are you?
-----BEGIN PGP SIGNATURE-----
Version: GnuPG v1

iQJ8BAEBCgBmBQJUVjYhXxSAAAAAAC4AKGlzc3Vlci1mcHJAbm90YXRpb25zLm9w
...
Ila46dxeh/DCOzAXdn9jWPtdyQGl/tk4qYPCT33oEvD1XrBWg6TuyInpz01rrDbZ
LFHCDQ4UojP+91j7MJ0w
=3Ki/
-----END PGP SIGNATURE-----
```

Email Signing



Good signature from [redacted]
Key ID: [redacted] / Signed on: Monday 20 October 2014 11:04 PM

From [redacted]

Subject **Re: Tomorrows meet**

Monday 20 October 2014 11:04 PM

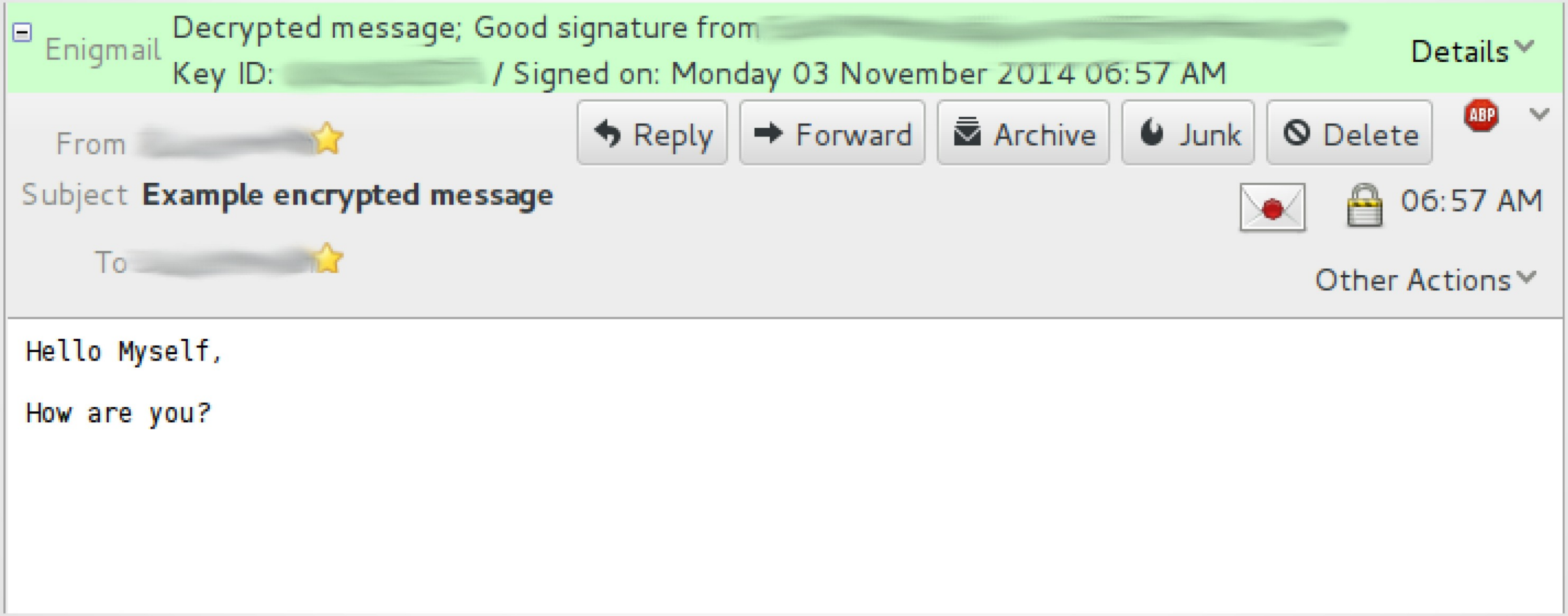
To [redacted] **4 more**

```
On Monday 20 October 2014 06:34 PM, [redacted] wrote:
[redacted] can only make it at 11.30.  Pl check with him.

I can come over tomorrow at 10.00.

--
[redacted]
```

Email Encryption

```
-----BEGIN PGP MESSAGE-----
Version: GnuPG v1

hQIMA/UHeoVMHUtXARAAnnncEg+jvlAvLMe7TtKaeCrlxdlNcbc3CXlJlddw0hho
ddDaj3njs+DhYYAd6AoPTQxEXXGttpb6uBGds0Fj4fg29DKjvEcDKgA5ognPVV8Q
jRunElrCVjRfiNiVtvJmF0W2a+37hTb+HcZaP4E/Zk3XT10kPDjiRmJ6cCBr7eUf
...
HoVK0fLNmiL3zcViosXkDAzvbKbODCZhhWHNgPIdUj5Idjwux86OWlbbZAZsbXuP
Th56R1MQEwNEuQ==
=Lt/6
-----END PGP MESSAGE-----
```

Email Encryption



Enigmail Decrypted message; Good signature from [redacted]
Key ID: [redacted] / Signed on: Monday 03 November 2014 06:57 AM

From [redacted]

Subject **Example encrypted message**

To [redacted]

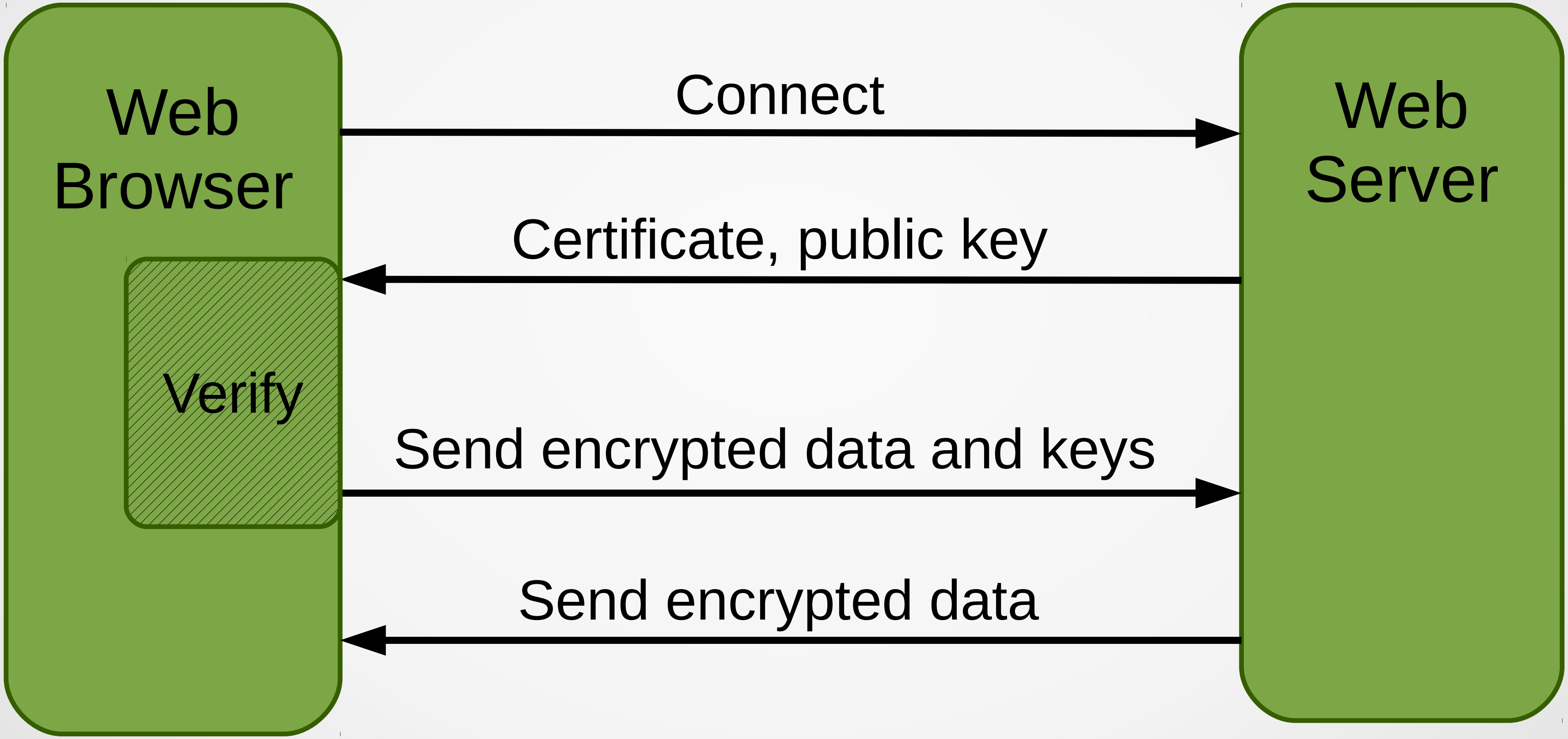
06:57 AM

```
Hello Myself,

How are you?
```

SSH Server Verification

- Ask administrator of the SSH server for fingerprint
- SSH client shows fingerprint on first connect
- Match the two and accept
- Fingerprint is stored and checked every time you connect

HTTPS



Web Browser

Web Server

Connect

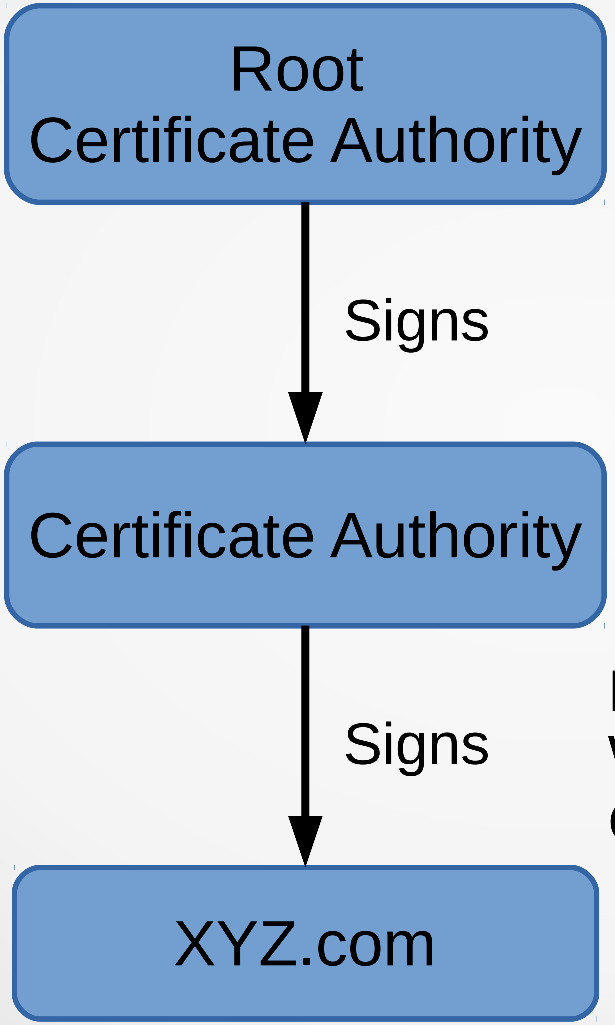
Certificate, public key

Verify

Send encrypted data and keys

Send encrypted data

HTTPS Public Key Verification



Public Key is hard-coded
into all Browsers

Public Key
Website Domain Name
Organisation Name (Optional)

References

- Applied Cryptography, 2nd edition – Bruce Schneier
- GNU Privacy Guard Manual – https://www.gnupg.org/documentation/manuals.html
- Books on Cryptography – https://en.wikipedia.org/wiki/Books_on_cryptography