Applied Cryptography

Hash Functions

Learning Objectives

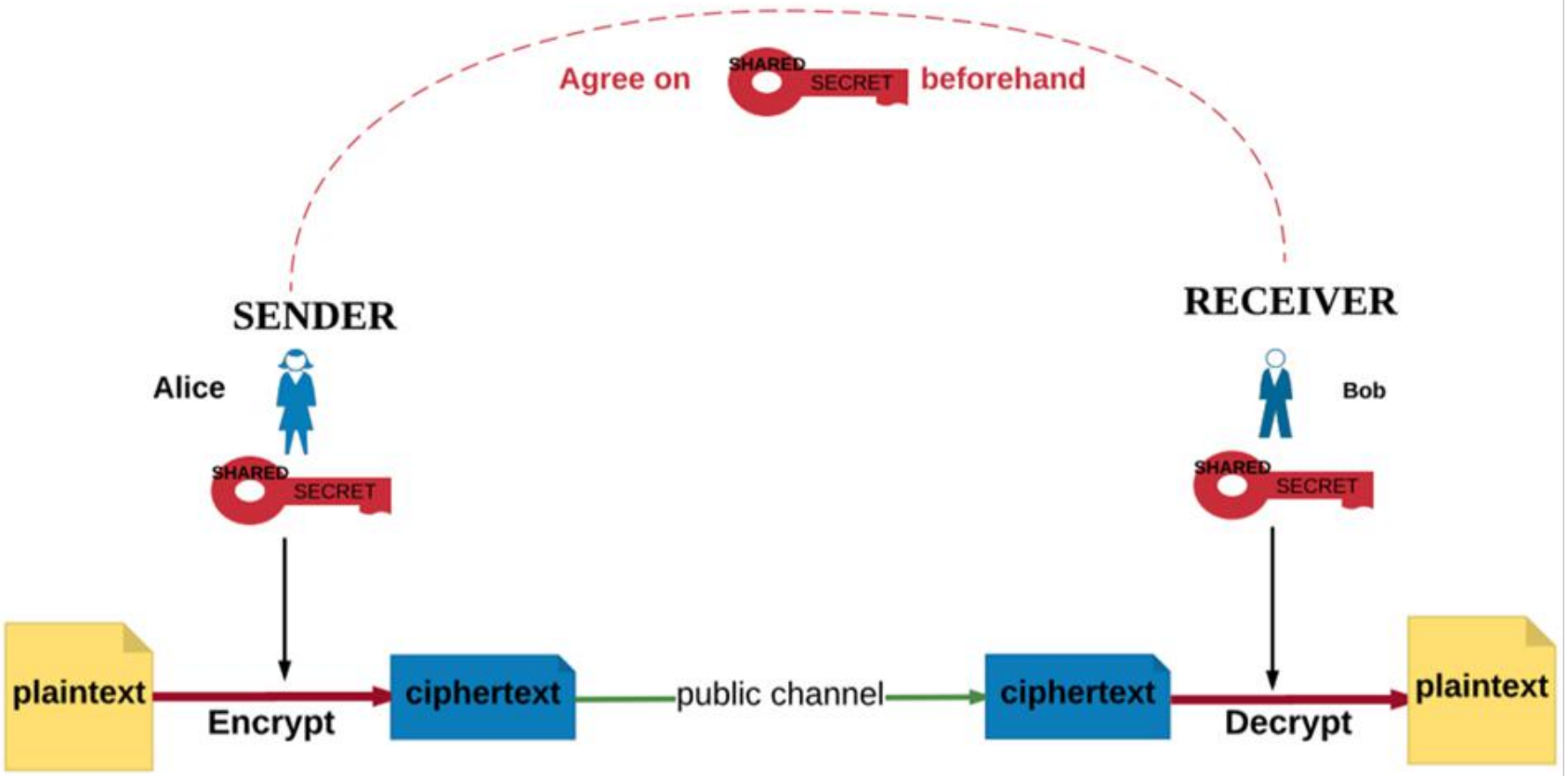
Upon completion of this unit:

- Students will be able to explain the use of hash functions in securing information.
- Students will be able to describe the basic requirements for a cryptographic hash function.
- Students will be able to identify commonly used algorithms for hashing.
- Students will be able to illustrate how hashing works using online tools.

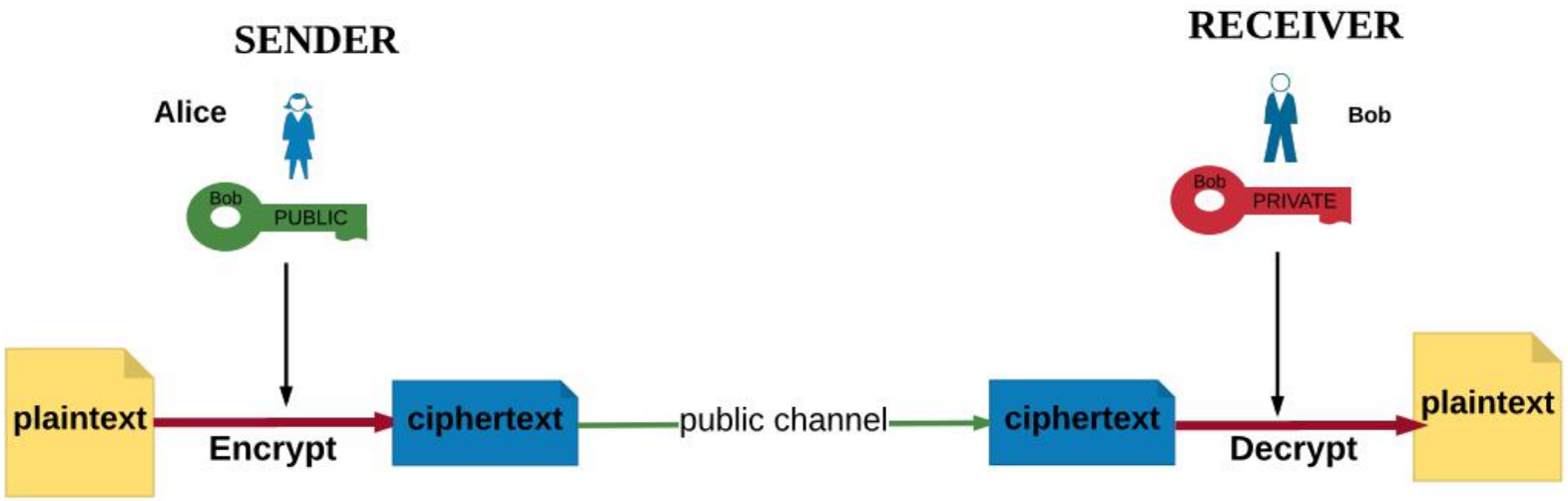
Recall: Cryptography and the Main Tenets of Information Security

- Confidentiality
  - Encryption algorithms
- Integrity
  - Hash functions
- Authenticity
  - Digital signatures, digital certificates
- Non-repudiation
  - Digital signatures, digital certificates

Recall: Symmetric Key Cryptography



Recall: Asymmetric Encryption



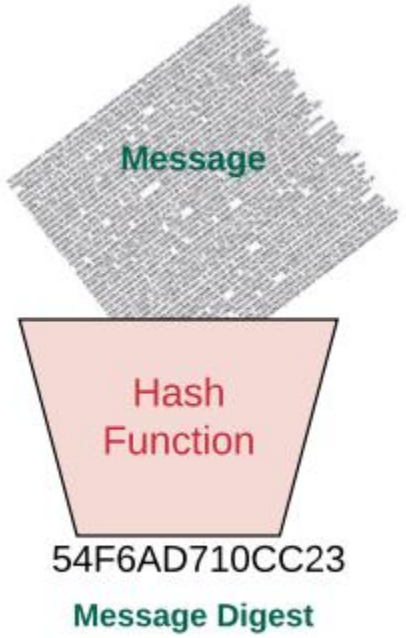
Note that no communication is necessary beforehand to agree on a shared secret key.

Recall: Problem of Integrity

- How does Bob know that the message was not altered on the way (in transit)?
- Solution: Hash functions
- Hash functions can also be used to check if data has altered at rest (when stored)

Hash Functions

- A hash function maps digital data of arbitrary size to digital data of **fixed size**. The hash is sometimes called a **message digest.**
- A cryptographic hash function is a hash function that is considered practically impossible to invert (one-wayness) or find collisions (i.e. two messages with the same hash value).



"Hash Function" by Yesem Kurt-Peker licensed under CC BY 4.0

Requirements for a Cryptographic Hash Function

- **Variable input/ Fixed output size:** Can be applied to data of practically any size but the output is always a fixed number of bits.
- **Pre-image resistant:** Computationally infeasible to find the input value for a given hash value (one-wayness).
- **Collision resistant:** Computationally infeasible to find two inputs that have the same hash value.
- **Efficiency:** Easy (fast) to compute so practical on hardware and software
- **Pseudorandomness:** The outputs pass tests designed to detect pseudorandomness.

Uses of Hash Functions

- **Digital Signatures:** When you sign messages digitally, the hash value of the message is encrypted instead of the message itself. This allows messages of arbitrary lengths to be signed.
- **Password Files:** Hashes of passwords are stored, not the passwords themselves. Because no one can see the plain password, this provides an extra layer of security in case the password file is stolen.

Uses of Hash Functions (continued)

- **Intrusion/Virus Detection:** A change in the hash value of a file may indicate an intrusion or a virus.
- **Construction of a pseudorandom number generator:** One of the required properties of a cryptographic function is that the output has to pass pseudorandomness tests.
- **File synchronization:** Whether to upload a file or not for synchronization (for example with the cloud storage) can be determined by checking the hash value of the file has changed or not since the last update

Examples of Cryptographic Hash Functions

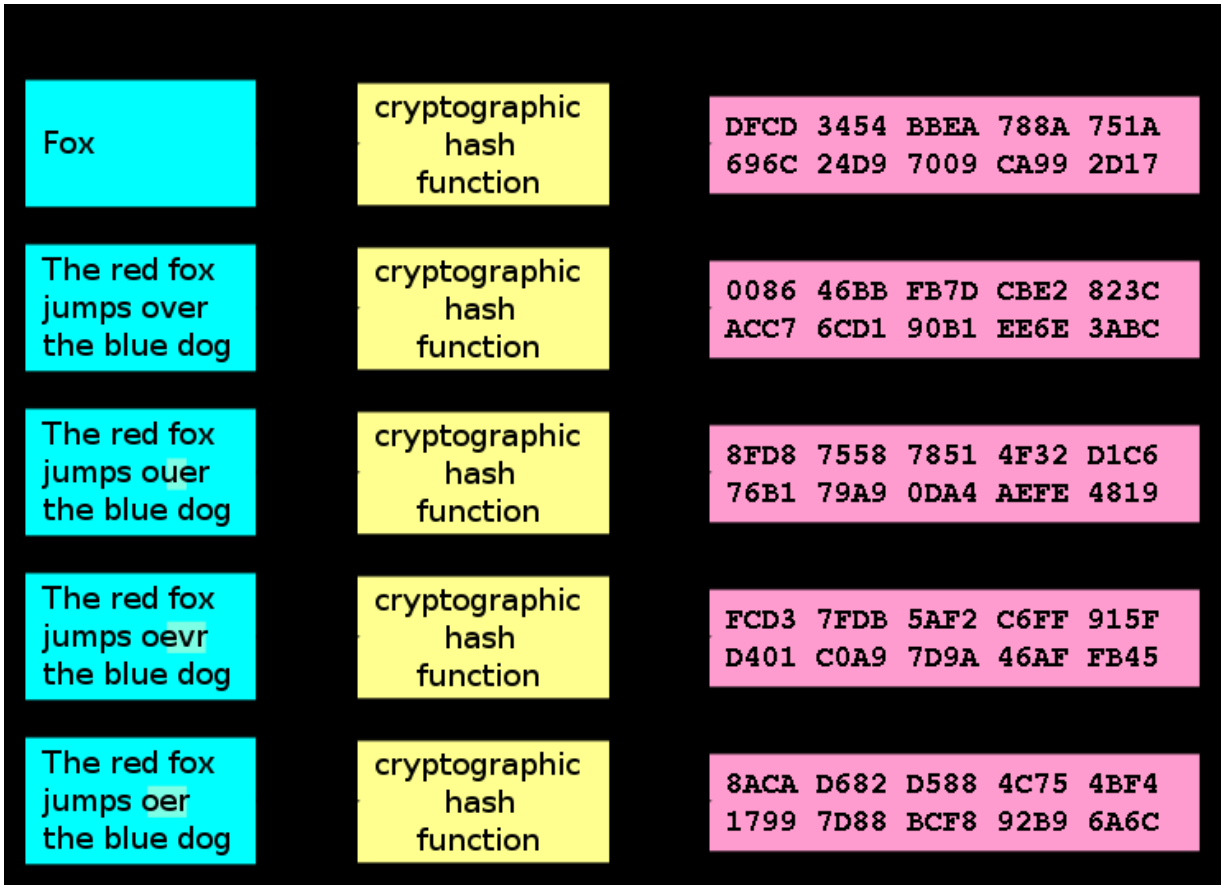
- SHA: Secure Hash Algorithm
  - SHA-1
    - Designed by NSA
    - Published by NIST in 1993 as Federal Info. Processing Standard
    - Theoretical attacks developed for SHA1 in 2005 suggested the algorithm may not be secure enough for ongoing use.
  - SHA-2
    - Also designed by NSA
    - Published by NIST in 2001 as Federal Info. Processing Standard
    - Includes six hash functions with digests that are 224, 256, 384 or 512 bits.

Examples of Cryptographic Hash Functions (continued)

- SHA-3
  - Designed by Bertoni, Daemen,Peeters, Van Assche
  - Published by NIST in 2015 as the new standard
  - Not meant to replace SHA-2 as SHA-2 has not been broken

- MD - Message Digest Algorithms
  - MD4
    - Designed by Rivest in 1990
    - 128 bit digests; used in TLS certificates
    - Practical collision attacks were developed against it
  - MD5
    - Similar to MD4; security severely compromised, so not suitable for cryptographic use.

A Cryptographic Hash Function (SHA-1) at Work



"Cryptographic Hash Function" by Jorge Stolfi derived from "Hash_function" by Helix84. Public Domain

Note that a small change in the input (in the word "over") drastically changes the output (digest). This is the so-called *avalanche effect*.

Hands-on: Comparing Hashes of Some Strings

- Go to the site http://www.fileformat.info/tool/hash.htm.
- Write a string in the string textbox and hit the hash button.
- Scroll down to see the hash though various algorithms.
- Copy and paste the hash values in a Word file.
- Make a small change in the string and hit the hash button again.
- Compare the hash values with those from the previous string.

Hands-on: Comparing Hashes of Some Strings (continued)

- Note how quickly all the hashes were calculated.
- Note that a small change in input gives a drastically different hash value.
- Choose a file from your computer and calculate the hash of it on this site.

Catalyzing Computing and Cybersecurity in Community Colleges

is funded by a National Science Foundation grant and is located at Whatcom Community College

237 West Kellogg Road
Bellingham, WA 98226

www.C5colleges.org