CSC3631 Cryptography

Introduction

Changyu Dong

Newcastle University

About me

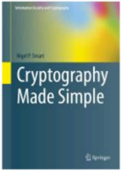
- Room 6-016, USB
- Email: changyu.dong@ncl.ac.uk

In the module

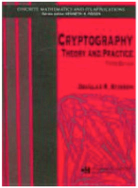
- Symmetric cryptography
    - Classical ciphers
    - Stream cipher
    - Block cipher
    - Hash function
    - Message authentication code
- Asymmetric cryptography

Reference books (Essential)

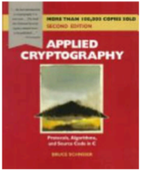
Cryptography Made Simple
(Nigel Smart, 2015)
-- online access through library



Cryptography: Theory and Practice
(Doug Stinson, 2006)

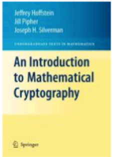


Applied Cryptography
(Bruce Schneier, 1996)
-- online access through library

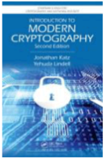

MORE THAN 100,000 COPIES SOLD
SECOND EDITION
APPLIED
CRYPTOGRAPHY
Protocols, Algorithms, and Source Code in C
BRUCE SCHNEIER

Reference books (Recommended)

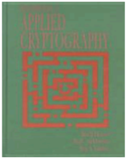
An introduction to mathematical cryptography
(Jeffrey Hoffstein, 2008)
-- online access through library



Introduction to Modern Cryptography
(Katz and Lindell, 2014)
-- online access through library



Handbook of applied cryptography
(Menezes et al., 2001)
-- http://cacr.uwaterloo.ca/hac/



Module Delivery (27 Sep - 4 Nov)

- Lectures Monday 11:30 and Tuesday 12:30
- You can find on Canvas
  - Slides and other materials
  - Short Videos
- Online Q & A session Thursday 17:30
  - `https://newcastleuniversity.zoom.us/j/89186001803`
  - Meeting ID: 891 8600 1803
  - Please book a slot through email before the session.

Coursework 1

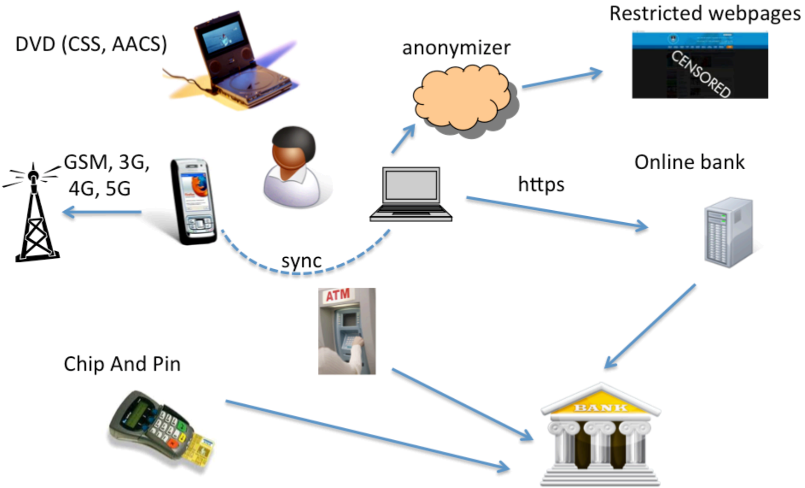
- 5 parts online quiz
- 20 marks in total
- multiple choices, filling-in-the-blanks, true or false ...
- 0.5 or 1 mark per question
- Only 1 attempt for each part
- You can choose to complete all 5 parts in one go, or do one each week as you progress.
- Deadline that you must finish all 5 parts by: Friday 12 Nov

Learn Cryptography in 20 minutes

- Symmetric cryptography
    - Symmetric encryption
        - Stream cipher
        - Block cipher
    - Hash function
    - Message authentication code
- Asymmetric cryptography
    - Public key encryption
    - Digital signature

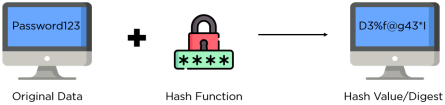
Symmetric Encryption

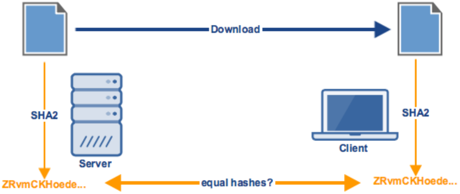
- Encrypt a message using a key, decrypt with the same key
- Stream cipher: encrypt/decrypt bit by bit
- Block cipher: encrypt/decrypt block by block
- Three algorithms:
    - Key generation: $Gen(n) \rightarrow k$
    - Encryption: $Enc_k(m) \rightarrow c$
    - Decryption: $Dec_k(c) \rightarrow m$
- Property 1: decryption can be done and can only be done with the same key used in encryption
    - $Dec_k(Enc_k(m)) = m$
    - For all $k' \neq k$, $Dec_{k'}(Enc_k(m)) \neq m$
- Property 2: (without the key) the ciphertext leaks no useful information about the plaintext
- Usage: to hide data from adversaries.

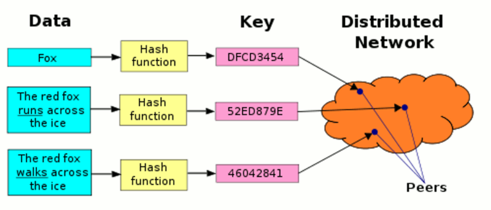

DVD (CSS, AACS)
anonymizer
Restricted webpages
CENSORED
GSM, 3G,
4G, 5G
https
Online bank
sync
ATM
Chip And Pin
BANK

Hash function

- Compress data of any size into a fixed length hash value (message digest).
  - $H(m) \rightarrow h$
- Property 1: Same data, same hash value (deterministic)
- Property 2: The hash value is the "fingerprint" of the original data (collision resistant)
  - The hash values are completely different even if only 1 bit is changed
- Property 3: From the hash value you cannot go back to the original data (one way).
- Usage 1: to hide data from adversaries (no decryption back)
- Usage 2: to authenticate data and detect modification
- Usage 3: fingerprinting data


Password123
Original Data
****
Hash Function
D3%f@g43*I
Hash Value/Digest


Download
SHA2
Server
Client
SHA2
ZRvmCKHoede...
equal hashes?
ZRvmCKHoede...

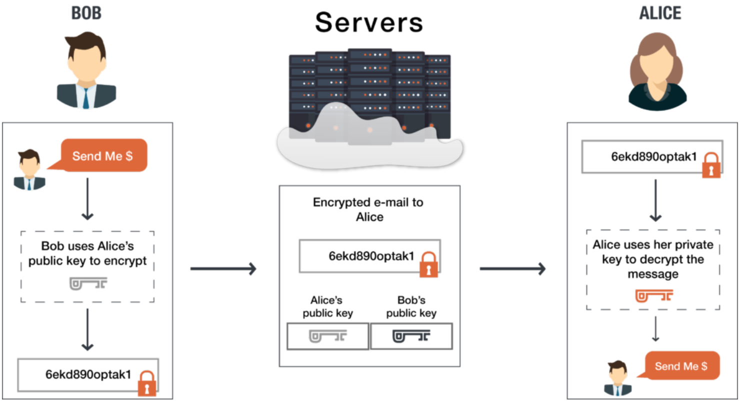

Data
Key
Distributed Network
Fox
Hash function
DFCD3454
The red fox runs across the ice
Hash function
52ED879E
The red fox walks across the ice
Hash function
46042841
Peers

Message Authentication Code

- Create a tag for a piece of data with a key, the tag can be verified using the same key to detect modification
- Three algorithms:
  - Key generation $Gen(n) \rightarrow k$
  - Mac tag generation: $Mac_k(m) \rightarrow t$
  - Tag verification: $Verify_k(t, m) \rightarrow 0 \textit{ or } 1$
- Property 1: only unmodified message can pass verification
  - $Verify_k(Mac_k(m), m) = 1$
  - $t = Mac_k(m)$, $m' \neq m$, $Verify_k(Mac_k(m), m') = 0$
- Property 2: An adversary cannot forge a tag for a message whose tag has not been seen by the adversary.
  - Because the adversary does not know the key
- Usage: to authenticate data and detect modification

Public key Encryption

- Encrypt a message using a key, which can be known by everyone (public key)
- Decrypt a ciphertext by a secret known by only one person (private key).
- The public key and private key are generated together
- Three algorithms:
  - Key generation: $Gen(n) \rightarrow (pk, sk)$
  - Encryption: $Enc_{pk}(m) \rightarrow c$
  - Decryption: $Dec_{sk}(c) \rightarrow m$
- Property 1: The ciphertext can only be decrypted using the correct private key.
  - For all $(pk, sk) \leftarrow Gen(n)$, $Dec_{sk}(Enc_{pk}(m)) = m$
  - For all $(pk, sk) \leftarrow Gen(n)$, $sk' \neq sk \; Dec_{sk'}(Enc_{pk}(m)) \neq m$
- Property 2: the ciphertext leaks no useful information about the plaintext (without the private key)
- Usage: to hide data from adversaries.



BOB

Send Me $

Bob uses Alice's public key to encrypt

6ekd890optak1

Servers

Encrypted e-mail to Alice

6ekd890optak1

Alice's public key

Bob's public key

ALICE

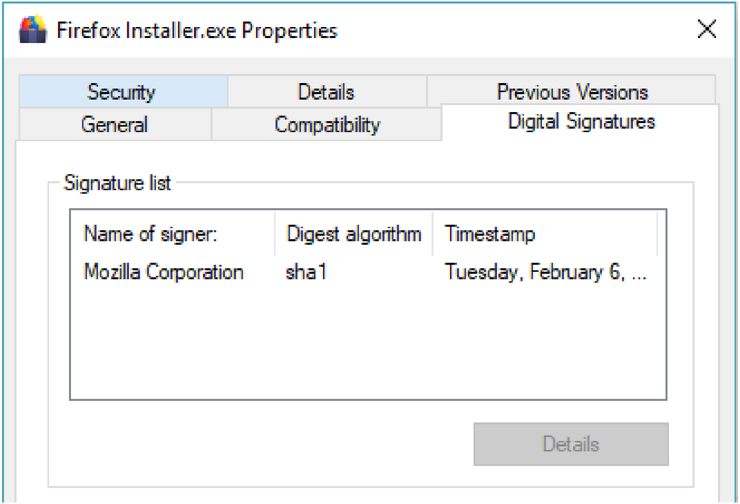
6ekd890optak1

Alice uses her private key to decrypt the message

Send Me $

Digital Signature

- Create a signature for a piece of data with a secret private key; the signature can be verified using the public key to detect modification
- Three algorithms:
  - Key generation $Gen(n) \rightarrow (pk, sk)$
  - Signature generation: $Sign_{sk}(m) \rightarrow \sigma$
  - Signature verification: $Verify_{pk}(\sigma, m) \rightarrow 0 \text{ } or \text{ } 1$
- Property 1: only unmodified message can pass verification
  - $Verify_{pk}(Sign_{sk}(m), m) = 1$
  - $\sigma = Sign_{sk}(m)$, $m' \neq m$, $Verify_{pk}(Sign_{sk}(m), m') = 0$
- Property 2: An adversary cannot forge a signature for a message whose signature has not been seen by the adversary.
- Property 3: If a signature is valid, the signer cannot deny it was generated by him/her.
- Usage: to authenticate data and detect modification



Firefox Installer.exe Properties

Signature list

Name of signer:	Digest algorithm	Timestamp
Mozilla Corporation	sha1	Tuesday, February 6, ...

Symmetric vs Asymmetric Cryptography

- Problems of Symmetric key cryptography
  - Require sharing a key: too many keys to manage if many users
  - Key must be distributed securely
  - No non-repudiation
- Problems of Asymmetric key cryptography
  - 100 - 1000 times slower than symmetric schemes
  - Keys are longer
  - Security are based on assumed hard problems (many are vulnerable to quantum computing)