DFSC 1316: digital forensic and information assurance fundamentals I

9. Cryptography Fundamentals

*Reference book: *Network Security Private Communication in a Public World by C Kaufman*

Motivations

- Why use cryptography?
  - To be (or at least to feel like) secure.
- Define “secure”?
  - Confidentiality.
  - Integrity.
  - Availability.
- How it relates to DF and IA?
  - Cryptography lays the foundation for modern information assurance.
  - It is the prerequisite for most digital forensic analysis.

Terminologies

- Cryptography
  - Comes from Greek words *κρυπτο* (*hidden* or *secret*) and *γραφη* (*writing*).
  - An art of mangling information into apparent unintelligibility.
  - Enables information exchange between two participants in a way that prevents others from reading it.
- Plaintext (cleartext), ciphertext, encryption, and decryption.

Secret Keys or Secret Algorithms?

- How about create an algorithm that is not known to anybody?
  - How hard to make it secure?
  - How hard to make it last?
  - How applicable it is?
  - What is a real-world example?

Secret Keys or Secret Algorithms?

- What are the alternatives?
  - Is it better in terms of security?
  - Is it better in terms of long-lasting?
  - what is a real-world example?
  - Is there any disadvantage?

Secret Keys or Secret Algorithms?

- Key
  - Modern *civil* cryptography systems tend to involve both an algorithm and a secret value. The secret value is know as the *key*.

Computational Difficulty

- Computational Difficulty
  - Cryptography algorithm are not impossible to break.
  - In the worst case, one can just try out all possible keys to find out the correct one.
  - It's all about the cost it takes to break it.

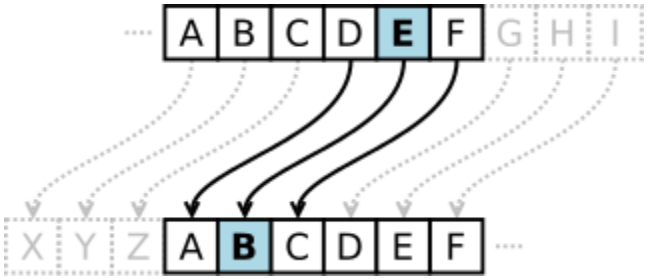
Key Size	Possible combinations
1-bit	2
2-bit	4
4-bit	16
8-bit	256
16-bit	65536
32-bit	4.2×10^{9}
56-bit (DES)	7.2×10^{16}
64-bit	1.8×10^{19}
128-bit (AES)	3.4×10^{38}
192-bit (AES)	6.2×10^{57}
256-bit (AES)	1.1×10^{77}

  - The link gives password recovery speeds:

    http://www.lockdown.co.uk/?pg=combi&s=articles

History and Simple Algorithms

- Caesar cipher
  - A substitution cipher.
  - Replace a letter with a fixed (the 3rd) number of the letter down the alphabet.



Plain: ABCDEFGHIJKLMNOPQRSTUVWXYZ
Cipher: XYZABCDEFGHIJKLMNOPQRSTUVW

Plaintext: THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG
Ciphertext: QEB NRFZH YOLTK CLU GRJMP LSBO QEB IXWV ALD

History and Simple Algorithms

- Captain Midnight Secret Decoder rings.
  - Caesar cipher with variable n.

History and Simple Algorithms

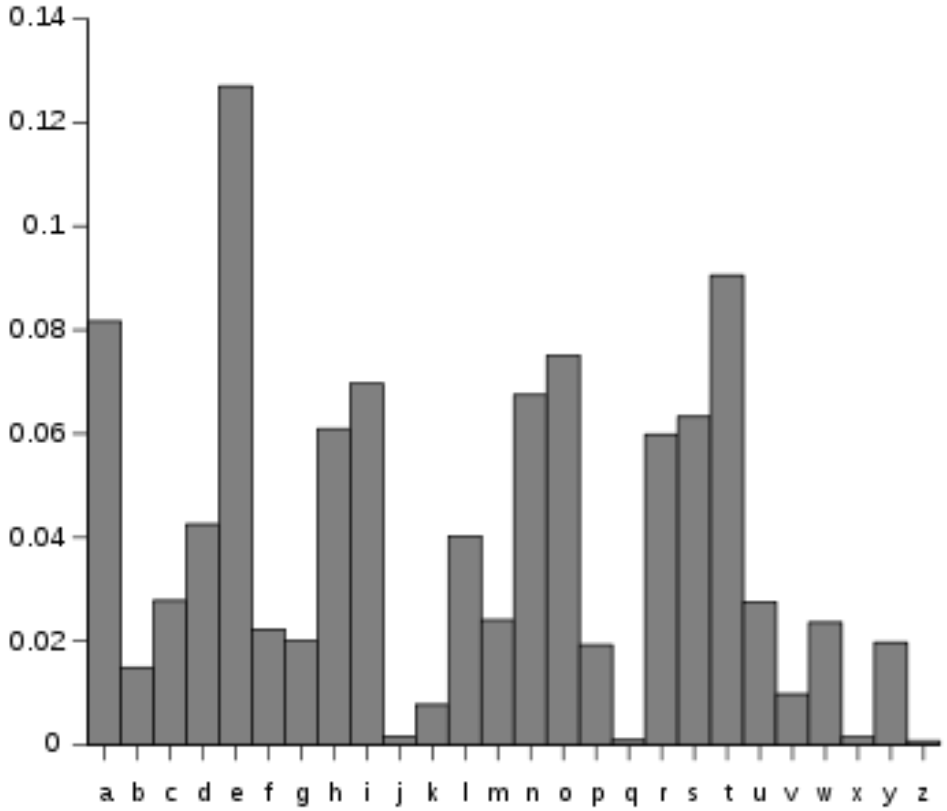
- Monoalphabetic cipher
  - An arbitrary map between a pair of letters.
  - How many combinations?
  - How to break?

- Can be broken with statistic knowledge
  - Try to decode this:

  cf lqr’xs xsnyctm n eqxxqgsy iqul qf wdcp eqqh, erl lqrx qgt iqul!

History and Simple Algorithms

- English language letter statistic



- Some online solvers: https://quipqiup.com/

Cryptanalysis: what is known

- Ciphertext Only
  - How hard to obtain?
  - How hard to analyze?

Cryptanalysis: what is known

- Known plaintext
  - How hard to obtain?
  - How hard to analyze?
  - Real-world example?
    - Encrypted ZIP.

Cryptanalysis: what is known

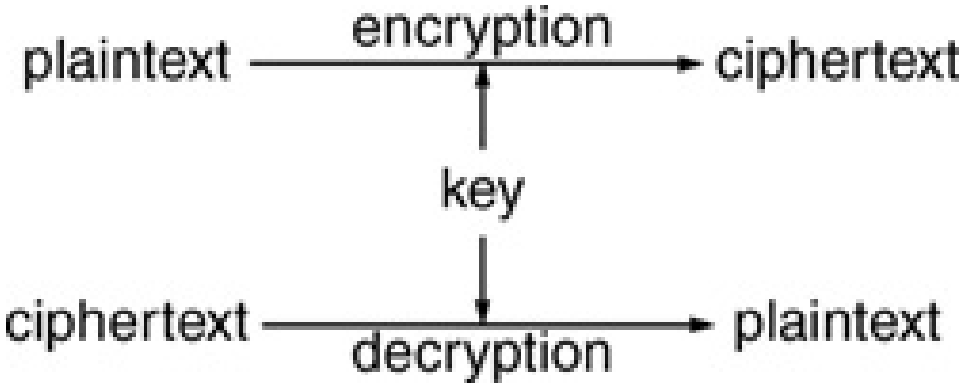
- Chosen plaintext
  - How hard to obtain?
  - How hard to analyze?
  - Real-world example?
    - The Battle of Midway.

Types of Cryptography

- Three types of cryptographic functions:
  - Hash functions: no key needed
  - Secret key (symmetric) cryptography: one key
  - Public key (asymmetric) cryptography: two keys – public and private

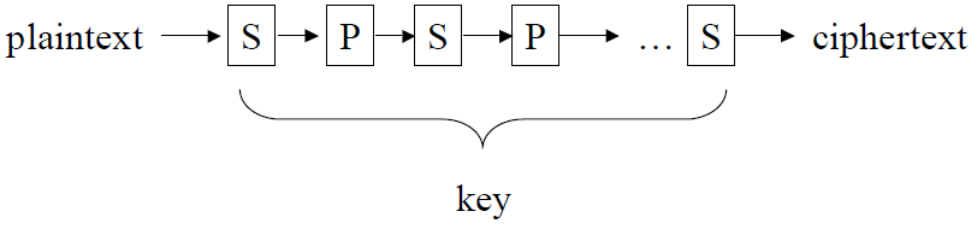
Secret Key Cryptography

- Also known as symmetric cryptography.
- Same key is used for both encryption and decryption.



Secret Key Cryptography

- Multiple application of interleaved substitutions and permutations.
- Ciphertext approximately has the same length as plaintext.



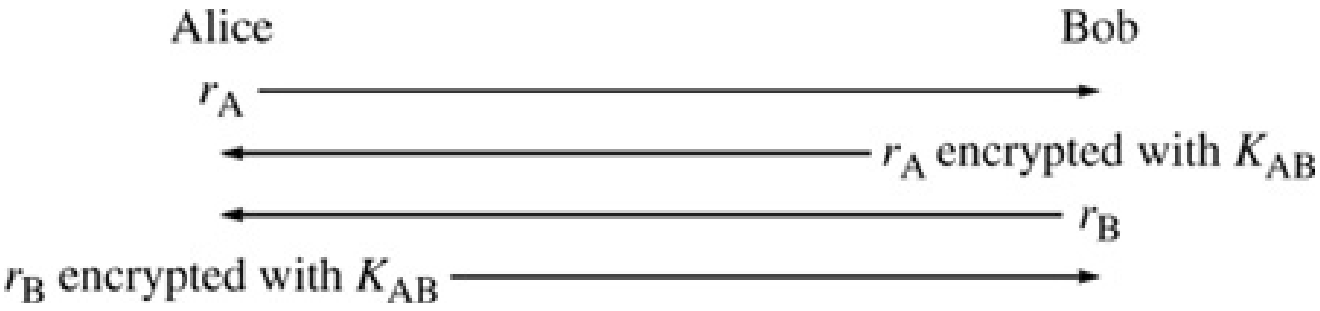
Application of Secret Key Cryptography

- Transmitting over an insecure channel
  - Applicable?
  - Any challenges?

- Secure storage on insecure media
  - Applicable?
  - Any challenges?

Application of Secret Key Cryptography

- Authentication
  - Strong authentication: prove knowledge of a secret without revealing it.



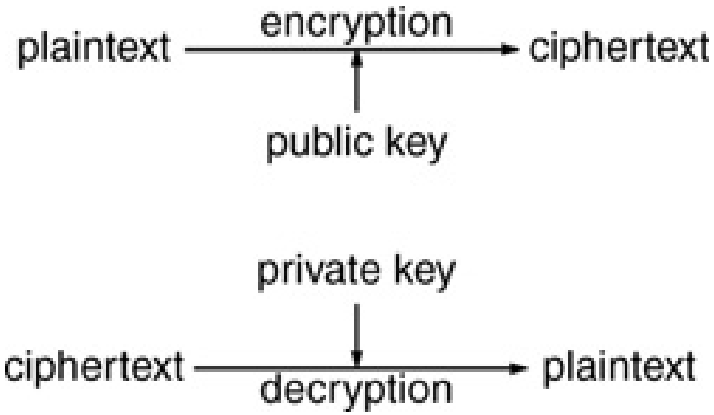
  - Must be secure enough to against chose plaintext attack.
  - How can it be compromised?

Public Key Cryptography

- An analogy

Public Key Cryptography

- A public/private key pair is used
  - Public key is known to public
  - Private key is to keep private
- Also known as Asymmetric cryptography.
- Much slower than secret key algorithms.



Application of Public Key Cryptography

- Transmission over an insecure channel
  - Applicable?
  - Any challenge?

- Secure storage on insecure media
  - Applicable?
  - Any challenge?

Application of Public Key Cryptography

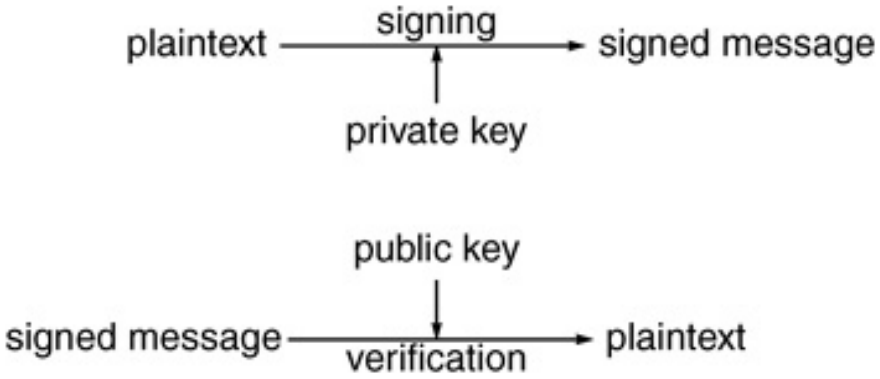
- Authentication
  - In secret key scheme, Alice will need remember as many keys as the person she wants to authenticate.
  - In public key scheme, Alice will only need to keep her private key secret. Public keys are publicly available.

Application of Public Key Cryptography

- Key exchange
  - Alice generate a secret key, encrypted it with Bob's public key.
  - Bob and Alice can then communicate with the secret key.

Application of Public Key Cryptography

- Digital Signature
  - Bob encrypt the message, or the Hash of the message, with his own private key.
  - Every one can verify the message because only Bob can encrypt the message.
  - Bob can not deny that he has signed (encrypted) the message because only he has the private key.

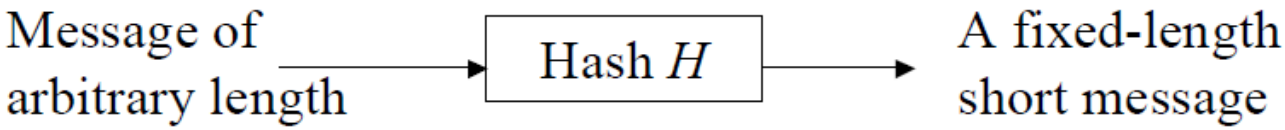


Application of Public Key Cryptography

- Can secret key algorithm be used for digital signature?
  - Alice and Bob share a secret key.
  - Alice sent Bob a message, but want to deny it.
  - Can she? How?

Hash Functions

- Also known as
  - Message digestion.
  - One-way transformation.
  - One-way functions.
- H(m) has fixed length, typically 128 or 160 bits.



Hash Functions

- Desired properties
  - Performance: easy to compute (fast).
  - One-way property: Given H(m) but not m, it is difficult to find m.
  - Weak collision free: Given H(m), it is difficult to find m', such that H(m) = H(m').
  - Strong collision free: Computationally infeasible to find m_1 and m_2, such that $H(m_1) = H(m_2)$.

Application of Hash Functions

- Transmission over an insecure channel
  - Applicable?
  - Any challenge?

- Secure storage on insecure media
  - Applicable?
  - Any challenge?

Application of Hash Functions

- Password hashing
  - The server has to authenticate a user by comparing his/her input password with local record.
  - How about store users' password on the local machine?

Application of Hash Functions

- Password hashing
  - In practice, instead of storing the password in cleartext, the server store the Hashed value of the password.
    - User input will first be Hashed, and then compared with local record.
  - Is it secure now?
  - What can potentially go wrong?
    - Guessable password.
    - Hash reverse.

Application of Hash Functions

- Message integrity
  - Hash can be used to generate a Message Authentication Code (MAC) to protect the integrity of messages.
    - Alice send Bob the message, and attach a Hash H(msg) alone with the message.
    - Bob will calculate the Hash again, and compare with the Hash he received.
    - Will it indicate the message has not been changed if it is a match?
  - Keyed Hash
    - Alice and Bob have some shared secret, say, a Key.
    - Instead of attach H(msg), Alice will attach with H(msg|Key).
    - Outsiders who don't know the Kay can't generate the Hash.

Application of Hash Functions

- Message Fingerprint
  - Used to protect the integrity of the forensic copy.
  - To know that your copy is exactly the same as the original copy
    - Compare the whole data
    - Compare the hash ← much easier

Logical Operations

- Binary numbers are commonly used to represent two states of an event.
  - 0 represents false, e.g., the comparison (5 > 10) is not true.
  - 1 represents true.

- Operations conducted on these logical concepts are called logical operations.

Logical Operations: NOT

- NOT is an unary operator.
  - An NOT operator is usually represented with ! or ~
  - It “flips” the value.
  - Truth table:

a	NOT a
0	1
1	0

Logical Operations: AND

- AND is a binary operator.
  - It is usually represented by &
  - It means “both”
  - Truth table:

a	b	a & b
0	0	0
0	1	0
1	0	0
1	1	1

Logical Operations: OR

- OR is a binary operator.
  - It is usually represented by |
  - It means “either”
  - Truth table:

a	b	a & b
0	0	0
0	1	1
1	0	1
1	1	1

Logical Operations: XOR

- XOR is a binary operator.
  - It is usually represented by $\oplus$
  - It indicates “same” or “different”
  - Truth table:

a	b	a & b
0	0	0
0	1	1
1	0	1
1	1	0

Exercise

- Logical operations are more commonly applied on a string of binary bits. It operates on each bits with no carry or borrow.
- Calculate the following:

1. 1001 & 0011

2. 1001 | 0011

3. 1001 $\oplus$ 0011