Public-key Cryptography

What Is Cryptography?

- Cryptography -- from the Greek for “secret writing” -- is the mathematical “scrambling” of data so that only someone with the necessary ***key*** can “unscramble” it.
- Cryptography allows secure transmission of private information over insecure channels (for example packet-switched networks).
- Cryptography also allows secure storage of sensitive data on any computer.

Basic Terminologies

- Cryptography deals with creating documents that can be shared secretly over public communication channels
- Cryptographic documents are decrypted with the key associated with encryption, with the knowledge of the encryptor
- The word cryptography comes from the Greek words: Krypto (secret) and graphein (write)
- Cryptanalysis deals with finding the encryption key without the knowledge of the encryptor
- Cryptology deals with cryptography and cryptanalysis
- Cryptosystems are computer systems used to encrypt data for secure transmission and storage

Basic Terminologies

- Keys are rules used in algorithms to convert a document into a secret document
- Keys are of two types:
  - Symmetric
  - Asymmetric
- A key is symmetric if the same key is used both for encryption and decryption
- A key is asymmetric if different keys are used for encryption and decryption

Basic Terminologies

- Examples:
  - Symmetric key methods
    - DES 56-bit
    - Triple DES 128-bit
    - AES 128-bit and higher
    - Blowfish 128-bit and higher
  - Asymmetric key methods
    - RSA (Rivest-Shamir-Adleman of MIT)
    - PGP (Phil Zimmerman of MIT)

Basic Terminologies

- Plaintext is text that is in readable form
- Ciphertext results from plaintext by applying the encryption key
- Notations:
  - M message, C ciphertext, E encryption, D decryption, k key
  - $E(M) = C$
  - $E(M, k) = C$
- Fact: $D(C) = M$, $D(C, k) = M$

Basic Terminologies

- Steganography is the method of hiding secret messages in an ordinary document
- Steganography does not use encryption
- Steganography does not increase file size for hidden messages
- Example: select the bit patterns in pixel colors to hide the message

Basic Terminologies

- Hash functions generate a digest of the message
- Substitution cipher involves replacing an alphabet with another character of the same alphabet set
- Mono-alphabetic system uses a single alphabetic set for substitutions
- Poly-alphabetic system uses multiple alphabetic sets for substitutions
- Caesar cipher is a mono-alphabetic system in which each character is replaced by the third character in succession. Julius Caesar used this method of encryption.

Basic Terminologies

- Vigenere cipher is an example of a poly-alphabetic cipher
- Vigenere cipher uses a 26 x 26 table of characters
- Vigenere method uses a keyword. Keyword repeated to fill length of plaintext. Each ciphertext character corresponds to the cell at the intersection of plaintext row and keyword column
- Vigenere method does not use repeated characters unlike Caesar cipher

Basic Terminologies

- Example of Vigenere cipher:
  ABCDEFGHIJ …
  BCDEFGHIJK …
  CDEFGHIJKL …
  DEFGHIJKLM …
  EFGHIJKLMN …

Plaintext:	BEAD
Keyword:	CABC
Ciphertext:	DABF

Basic Terminologies

- Hash algorithms take an arbitrary length message and create a fixed length digest known as Message Digest
- Well-known hash algorithms are MD-4 and MD-5
- Ron Rivest created the MD-x hash algorithms for NIST
- Block ciphers use blocks of text instead of single characters
- Electronic code book (ECB) uses plaintext blocks

Basic Terminologies

- ECB raises the possibility that identical blocks could generate identical ciphertext
- Cipher block chaining (CBC) uses a feedback loop
- In CBC, each plaintext block is XORed with the previous ciphertext block
- CBC eliminates identical blocks generating identical ciphertext

PKI

- Public Key Infrastructure (PKI) is a government initiative to protect computer systems
- Developed in the 1970s but has not been widely accepted. However, parts of the system are in extensive use today. These are Digital Certificates and Digital Signatures.
- Digital Certificates are given by trusted third parties, known as Certificate Authorities (CAs). Verisign (an offshoot of RSA) is a CA. Any organization can be a CA as long as there are people willing to believe their assessment of authenticity.

Digital Certificates

- Issued by trusted third parties known as Certificate Authorities (CAs)
- Verisign is a trusted third party
- Used to authenticate an individual or an organization
- Digital Certificates are usually given for a period of one year
- They can be revoked
- It is given at various security levels. Higher the security level, the CA verifies the authenticity of the certificate seeker more.

Digital Certificates

- Digital Certificates can be issued by any one as long as there are people willing to believe them

Digital Certificates

- Digital Certificates are part of the authentication mechanism. The other part is Digital Signature.
- When a user uses the digital signature, the user starts with their private key and encrypts the message and sends it. The receiver uses the sender's public key and decrypts the message
- In traditional encryption, the sender uses the public key of the receiver and encrypts the message and sends it and the receiver decrypts the message with their private key

Digital Certificates

- Additional authentication means used by CAs are:
  - Security token
  - Passive token
  - Active token
  - One time password

Digital Certificates

- Security token is usually a hardware device such as a Smart Card
- If the security token is a software token, it is usually associated with a particular workstation
- Security tokens use two-factor authentication using a password and a device (or an appropriate hardware identifier)

Digital Certificates

- Passive token is a storage device that holds multiple keys. Appropriate key is transmitted using the transmission device used.
- Inexpensive to manufacture
- Sometimes an extra PIN is required to use the passive token
- Examples:
  - Garage door opener
  - ATM card

Digital Certificates

- An Active token does not transmit any data, unlike a passive token
- Active tokens create another form of the base key (such as one-time password) or an encrypted form of the base key
- Smart cards are commonly used for active tokens

Digital Certificates

- A One-time password has a limited duration validity on a single use
- Generated using a counter-based token or a clock-based token
- Counter-based token is an active token that generates a one-time password based on a counter in the server and the secret key of the user
- Clock-based token is an active token that generates one-time passwords based on the server clock

PGP

- Developed by Phil Zimmerman at MIT
- Provides 256-bit encryption key
- Widely used for encrypting files such as email
- Message is first compressed
- A session key is created
- The compressed message is encrypted using the session key

PGP

- Session key alone is encrypted using the recipient’s public key
- The encrypted message and the encrypted session key are then sent to the receiver
- Receiver uses the private key to decrypt the session key first. Then the message is decrypted in a symmetric key way.

PGP

- PGP supports the following encryption methods:
  - CAST (named after the developers Carlisle Adams and Stafford Tavares) is owned by Nortel. It uses a 128-bit key. Freeware.
  - IDEA (International Data Encryption Algorithm). Not a freeware. Uses 128-bit key
  - Triple DES. Freeware. Uses three 56-bit keys

Classical Cryptography: Secret-Key or Symmetric Cryptography

- Alice and Bob agree on an encryption method and a shared ***key***.
- Alice uses the key and the encryption method to ***encrypt*** (or ***encipher***) a message and sends it to Bob.
- Bob uses the same key and the related decryption method to ***decrypt*** (or ***decipher***) the message.

Advantages of Classical Cryptography

- There are some very fast classical encryption (and decryption) algorithms
- Since the speed of a method varies with the length of the key, faster algorithms allow one to use longer key values.
- Larger key values make it harder to guess the key value -- and break the code -- by brute force.

Disadvantages of Classical Cryptography

- ***Requires secure transmission of key value***
- Requires a separate key for each group of people that wishes to exchange encrypted messages (readable by any group member)
  - For example, to have a separate key for each pair of people, 100 people would need 4950 different keys.

Public-Key Cryptography: Asymmetric Cryptography

- Alice generates a key value (usually a number or pair of related numbers) which she makes public.
- Alice uses her public key (and some additional information) to determine a second key (her ***private key***).
- Alice keeps her private key (and the additional information she used to construct it) secret.

Public-Key Cryptography (continued)

- Bob (or Carol, or anyone else) can use Alice’s public key to encrypt a message for Alice.
- Alice can use her private key to decrypt this message.
- No-one without access to Alice’s private key (or the information used to construct it) can easily decrypt the message.

An Example: Internet Commerce

- Bob wants to use his credit card to buy some brownies from Alice over the Internet.
- Alice sends her public key to Bob.
- Bob uses this key to encrypt his credit-card number and sends the encrypted number to Alice.
- Alice uses her private key to decrypt this message (and get Bob's credit-card number).

Hybrid Encryption Systems

- All known public key encryption algorithms are much slower than the fastest secret-key algorithms.
- In a ***hybrid*** system, Alice uses Bob's public key to send him a secret shared ***session key***.
- Alice and Bob use the session key to exchange information.

Internet Commerce (continued)

- Bob wants to order brownies from Alice and keep the ***entire transaction*** private.
- Bob sends Alice his public key.
- Alice generates a session key, encrypts it using Bob’s public key, and sends it to Bob.
- Bob uses the session key (and an agreed-upon symmetric encryption algorithm) to encrypt his order, and sends it to Alice.

Digital Signatures: Signing a Document

- Alice applies a (publicly known) ***hash function*** to a document that she wishes to "sign." This function produces a ***digest*** of the document (usually a number).
- Alice then uses her ***private*** key to "encrypt" the digest.
- She can then send, or even broadcast, the document with the encrypted digest.

Digital Signature Verification

- Bob uses Alice’s ***public*** key to “decrypt” the digest that Alice “encrypted” with her private key.
- Bob applies the hash function to the document to obtain the digest directly.
- Bob compares these two values for the digest. If they match, it proves that Alice signed the document and that no one else has altered it.

Secure Transmission of Digitally Signed Documents

- Alice uses her ***private*** key to digitally sign a document. She then uses Bob's ***public*** key to encrypt this digitally signed document.
- Bob uses his ***private*** key to decrypt the document. The result is Alice's digitally signed document.
- Bob uses Alice's ***public*** key to verify Alice's digital signature.

Historical Background

- 1976: W. Diffie and M.E. Hellman proposed the first public-key encryption algorithms -- actually an algorithm for public ***exchange*** of a secret key.
- 1978: L.M Adleman, R.L. Rivest and A. Shamir propose the RSA encryption method
  - Currently the most widely used
  - Basis for the spreadsheet used in the lab

The RSA Encryption Algorithm

- Use a random process to select two large prime numbers ***P*** and ***Q***. Compute the product ***M = P*Q***. This number is called the ***modulus***, and is made publicly available.
  - RSA currently recommends a modulus that's at least 768 bits long.
- Also compute the ***Euler totient***
  T = (P-1)*(Q-1). Keep this number (as well as ***P*** and ***Q***) secret.

RSA (continued)

- Randomly choose a public key ***E*** that has no factors in common with ***T = (P-1)*(Q-1)***.
- Compute a private key ***D*** so that ***E*D*** leaves a remainder of 1 when divided by ***T***.
  - We say ***E*D*** is ***congruent*** to 1 ***modulo T***
- Note that ***D*** is easy to compute only if one knows the value of T. This is essentially the same as knowing the values of P and Q.

RSA (continued)

- If ***N*** is any number that is not divisible by ***M***, then dividing N^{E*D} by ***M*** and taking the remainder yields the original value ***N***.
  - This is a relatively deep mathematical theorem, which we can write as $N^{E*D} \bmod M = N$.)
- If ***N*** is a numeric encoding of a block of plaintext, the cyphertext *is* $C = N^E \bmod M$.
- Then $C^D \bmod M = (N^E)^D \bmod M = N^{E*D} \bmod M = N$. Thus, we can recover the plaintext ***N*** with the private key ***D***.

Why RSA Works

- Multiplying P by Q is ***easy***: the number of operations depends on the ***number of bits*** (number of digits) in P and Q.
- For example, multiplying two 384-bit numbers takes approximately $384^2 = 147,456$ bit operations

Why RSA Works (2)

- If one knows only M, finding P and Q is ***hard***: in essence, the number of operations depends on the ***value*** of M.
    - The simplest method for factoring a 768-bit number takes about $2^{384} \approx$ 3.94 □10^{115} trial divisions.
    - A more sophisticated methods takes about $2^{85} \approx$ 3.87 □ 10^{25} trial divisions.
    - A still more sophisticated method takes about $2^{41} \approx$ 219,000,000,000 trial divisions

Why RSA Works (3)

- No-one has found an really quick algorithm for factoring a large number ***M***.
- No-one has proven that such a quick algorithm doesn't exist (or even that one is unlikely to exist).
- Peter Shor has devised a very fast factoring algorithm for a ***quantum computer***, if anyone manages to build one.