CS 3IS3 Fall 2007

03 Basic Cryptography

William M. Farmer

Department of Computing and Software
McMaster University

22 October 2007

What is Cryptography?

- **Definition 1**: Cryptography is the art and science of concealing meaning (Bishop).
- **Definition 2**: Cryptography is a collection of mathematical techniques for:
  - Protecting data confidentiality.
  - Protecting data integrity.
  - Verifying the identity of objects.
  - Verifying the identity of subjects.
  - Producing random objects.

Principal Cryptographic Techniques

- Conventional encryption.
- Cryptographic hashing.
- One-way encryption.
- Public key encryption.
- Random number generation.

Conventional Encryption

- A single key is required that is kept secret.
- Encryption: plaintext, key $\xrightarrow{f}$ ciphertext.
- Decryption: ciphertext, key $\xrightarrow{f^{-1}}$ plaintext.
- f and f^{-1} are the encryption and decryption algorithms, respectively.
- Main assumption: Computation of the plaintext from the ciphertext is mathematically infeasible without the key.
- In practice, the security of the process depends primarily on maintaining the secrecy of the key!

Cryptosystems

- A cryptosystem is a tuple $(\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ where:
  - $\mathcal{P}$ is a set of plaintexts.
  - $\mathcal{C}$ is a set of ciphertexts.
  - $\mathcal{K}$ is a set of keys.
  - $\mathcal{E}$ is a set of encryption functions $f : \mathcal{P} \times \mathcal{K} \to \mathcal{C}$.
  - $\mathcal{D}$ is a set of decryption functions $f^{-1} : \mathcal{C} \times \mathcal{K} \to \mathcal{P}$.
- Example: System of Caesar ciphers.

Ciphers

- A cipher is an encryption/decryption method.
- Mono-alphabetic ciphers (letter-for-letter substitution).
  - Caesar (rotation) ciphers (25 possible keys).
  - Shuffle ciphers (26! possible keys).
- Cipher techniques:
  - Transposition.
  - Substitution.
  - Stream translation.
  - Block translation.

Cryptanalysis

- Cryptanalysis is the process of discovering how to decrypt ciphertext without the secret key.
  - Uses mathematics and statistics.
- Approaches:
  - Brute force: try all possible keys.
  - Exploit known plaintext.
  - Exploit chosen plaintext.
  - Analyze encryption and decryption algorithms.
  - Exploit weaknesses in implementations (so-called side channel attacks).
- Criteria for measuring the effectiveness of a cipher:
  - Cost of breaking the cipher vs.
    Value of the encrypted information.
  - Time required to break the cipher vs.
    Useful lifetime of the encrypted information.

Data Encryption Standard (DES)

- For many years the most widely used conventional encryption algorithm.
  - Developed by IBM in the late 1960s.
  - Adopted by the USA National Institute of Standards and Technology (NIST) in 1977.
- Process:
  - Same algorithm used for encryption and decryption.
  - Encryption is performed in 64-bit blocks.
  - Change of single input bit changes almost all output bits.
  - Key is 56 bits long (as requested by USA National Security Agency (NSA)).
- Security concerns:
  - Key length (brute force attacks can now work in less than 24 hours),
  - Internal algorithm structure (design analysis is classified).

Advanced Encryption Standard (AES)

- Competitively selected replacement for DES.
  - Developed by Joan Daemen and Vincent Rijmen.
  - Adopted by the USA NIST in 2001.
  - Expected to be used worldwide.
- Process:
  - Same algorithm used for encryption and decryption.
  - Encryption is performed in 128-bit blocks.
  - Key is 128, 192, or 256 bits long.
  - AES algorithm is much faster than DES algorithm.
- Security issues:
  - AES was approved in 2003 by the USA NSA for Top Secret information when used with 192- or 256-bit keys.
  - As of 2006, the only successful attacks have been side channel attacks based on weaknesses in particular implementations of AES.
  - The algorithm is unclassified, publicly disclosed, and royalty-free.

International Data Encryption Algorithm (IDEA)

- Developed by Xuejia Lai and James Massey of Swiss Federal Institute of Technology and published in 1990.
  - Patented by Ascom-Tech AG.
  - No license fee required for noncommercial use.
- Process:
  - Same algorithm used for encryption and decryption.
  - 128-bit key is used to encrypt data in 64-bit blocks.
- Major alternative to DES before AES.
  - Faster than DES.
  - Considered much more secure than DES.
  - Included in the Pretty Good Privacy (PGP) package.

Blowfish

- Developed by Bruce Schneier around 1993.
  - Available without fee for all uses.
  - Intended as a general-purpose, public-domain replacement for DES.
- Fast, compact, easy to implement.
- Encrypts data in 64-bit blocks.
- Key length may be chosen between 32 and 448 bits.
  - Higher speed and higher security can be traded off.
- Considered to be an extremely strong algorithm.

Key Distribution Centers

- Main challenge for conventional encryption: Secret key distribution!
  - Often too many secret keys are needed to deliver them all physically.
- A key distribution center (KDC) holds a unique master key for each end system.
- Communication between end systems is encrypted using a temporary key called a session key.
  - One end system A requests a session key from KDC to communicate with another end system B.
  - The KDC sends A back a message encrypted with A's master key containing the session key and a message for B encrypted with B's master key.
  - The latter message, which contains the session key and A's identity, is sent to B by A.
- The whole system fails if the KDC is compromised.

Hashing

- Given an object as input, a hash function returns an identification code (called a hash code) for the object.
- A hash function has the following properties:
  - The output has a fixed size, much smaller than the size of the input.
  - The function is many-to-one (so collisions are possible).
  - The function is deterministic and easy to compute.
- Hash functions are used to:
  - Build rapidly accessible data storage structures called hash tables.
  - Produce checksums for checking data integrity.

Cryptographic Hashing

- A cryptographic hash function is a hash function whose purpose is to produce a “fingerprint” (called a message digest, cryptographic hash code, or cryptographic checksum) of an input object.
- A cryptographic hash function h has the following properties:
  - One-way property: Given a hash code c, it is mathematically infeasible to find an object x such that $h(x) = c$.
  - Weak collision property: Given an object x, it is mathematically infeasible to find another object y such that $h(x) = h(y)$.
  - Strong collision property: It is mathematically infeasible to find two objects x and y such that $h(x) = h(y)$.
- A keyed cryptographic hash function requires a cryptographic key when it is applied.

One-Way Encryption

- A one-way encryption function maps a plaintext to a ciphertext in such a way that it is mathematically infeasible to obtain the plaintext from the ciphertext.
  - No key is needed.
- Application: Password authentication.
  - When a password is declared, it is mapped by a one-way encryption function to ciphertext that is then stored on the system.
  - The plaintext is never stored.
  - A plaintext that is claimed to be a password is verified by comparing the ciphertext it produces with the ciphertext stored on the system.

Public Key Encryption

- Discovery:
  - Discovered but held secret by USA NSA and UK Communications-Electronic Security Group in mid to late 1960s.
  - Discovered and publicized by Whitfield Diffie and Martin Hellman at Stanford University in 1976.
- Motivation:
  - Difficulty of secret key distribution: secrecy must be shared.
  - Need for digital signatures that can be verified by arbitrary parties.

Public Key Encryption: Basic Process

- Each end system has two keys:
  - Private key that is kept secret.
  - Public key that is made public.
- Encryption: plaintext, public key $\xrightarrow{f}$ ciphertext.
- Decryption: ciphertext, private key $\xrightarrow{f}$ plaintext.
- Signature writing: plaintext, private key $\xrightarrow{f}$ ciphertext.
- Signature reading: ciphertext, public key $\xrightarrow{f}$ plaintext.
- The same algorithm is used for both encryption and decryption.
- It is mathematically infeasible to derive the private key from the public key.

Public Key Encryption Applications (1)

1. Confidentiality.
    - The sender encrypts the plaintext message with the receiver's public key.
    - The receiver decrypts the ciphertext message with its private key.
2. Integrity, digital signature, and nonrepudiation.
    - The sender encrypts the message digest of the sent text with its private key.
    - The receiver decrypts the encrypted message digest with the sender's public key and compares it with the message digest of the received text.

Public Key Encryption Applications (2)

3. Confidentiality and integrity.
    - The sender encrypts the plaintext message with its private key.
    - The sender encrypts the ciphertext message with the receiver's public key.
    - The receiver decrypts the ciphertext message with its private key.
    - The receiver decrypts the ciphertext message with the sender's public key.
4. Secret key exchange.

Diffie-Hellman Key Exchange Algorithm

- Appeared in original 1976 Diffie-Hellman paper.
- Used only for secret key exchange.

RSA Algorithm

- Developed by Ron Rivest, Adi Shamir, and Len Adleman at MIT in 1977.
- Supports confidentiality, digital signature, and secret key exchange.
- Most widely used public key algorithm.
- The keys are generated from two large prime numbers p and q.
  - p and q are private.
  - The product of p and q is public.
  - Underlying assumption: Factoring sufficiently large integers is assumed to be mathematically infeasible.
- RSA is believed to be secure if the keys are sufficiently long.
  - RSA keys are usually 1024–2048 bits long.

Key Management

- Key management is the part of cryptography concerned with the distribution of cryptographic keys.
- Key management is critical to the effective application of cryptographic methods. Due to its human component, it is perhaps the most challenging aspect of cryptography.
- Services provided by key management systems:
  - Key generation.
  - Subject identity and authentication.
  - Subject to key binding.
  - Key distribution.
  - Key revocation.

Session Keys

- A session key is a conventional encryption key that is used for a single communication session.
- Sessions key are discarded after the communication session ends.
- The use of session keys helps prevent:

  1. Attacks on the cipher by reducing the amount data that is encrypted and the time the key is in use.
  2. Replay attacks because the key is used only once.
  3. Forward searches because the key is used only once.

- Session key distribution is nontrivial because a session key must be distributed as secret data to the two different subjects who may not know each other.

Classical Session Key Exchange

- The basic classical protocol for Alice to request a session key from Cathy to communicate with Bob:
  1. Alice $\to$ Cathy : $\{\text{want session key for Bob}\}k_{\text{Alice,Cathy}}$
  2. Cathy $\to$ Alice : $\{k_{\text{session}}\}k_{\text{Alice,Cathy}} \parallel \{k_{\text{session}}\}k_{\text{Bob,Cathy}}$
  3. Alice $\to$ Bob : $\{k_{\text{session}}\}k_{\text{Bob,Cathy}}$
- Main assumption: Alice and Bob both trust Cathy.
- Problem: Bob does not know who is sending him messages encrypted with the session key.
  - This opens Bob up to replay attacks.
  - Bob needs a way to authenticate Alice.

Needham-Schroeder Protocol

- The Needham-Schroeder protocol for Alice to request a session key from Cathy to communicate with Bob:
  1. Alice $\rightarrow$ Cathy : $\{$Alice $\|$ Bob $\|$ random $n_1\}k_{\text{Alice,Cathy}}$
  2. Cathy $\rightarrow$ Alice : $\{$Alice $\|$ Bob $\|$ n_1 $\|$ k_{session} $\|$ $\{$Alice $\|$ $k_{\text{session}}\}k_{\text{Bob,Cathy}}\}k_{\text{Alice,Cathy}}$
  3. Alice $\rightarrow$ Bob : $\{$Alice $\|$ $k_{\text{session}}\}k_{\text{Bob,Cathy}}$
  4. Bob $\rightarrow$ Alice : $\{$random $n_2\}k_{\text{session}}$
  5. Alice $\rightarrow$ Bob : $\{f(n_2)\}k_{\text{session}}$
- Main assumption: Alice and Bob both trust Cathy.
- Problem: The session key, which is usually pseudorandomly generated, might be predicted.
  - Can be solved with the use of timestamps, but this requires clock synchronization.
  - Can also be solved with the use of a pseudorandomly generated session number.

Public Key Session Key Exchange

- Public key protocol for Alice to send Bob a session key for communication with him:
  1. Alice $\rightarrow$ Bob : $\{k_{\text{session}}\}e_{\text{Bob}}$
     where e_{Bob} is Bob's public key.
- Problem: Bob does not know who is sending him messages encrypted with his key.
- Better protocol:
  1. Alice $\rightarrow$ Bob : $\{\text{Alice} \,\|\, \{k_{\text{session}}\}d_{\text{Alice}}\}e_{\text{Bob}}$
     where d_{Alice} is Alice's private key.
- A man-in-the-middle attack can work against this protocol.
- Need a way of binding identity to a public key.

Certificates

- A certificate is a message that binds an identity to a cryptographic key (usually a public key).
- A certificate issued by Cathy that binds Alice to her public key has the form

$$\{e_{\text{Alice}} \parallel \text{Alice} \parallel T\}d_{\text{Cathy}}.$$

- A certificate authority (CA) is an entity that issues certificates.
- $X\langle\langle Y\rangle\rangle$ represent a certificate that the CA X issued for Y.

Certificates Signature Chains

- Certificates are used to validate the issuers of other certificates.
- A certification signature chain is a list of certificates of the form

$$X_1\langle\langle X_2\rangle\rangle X_2\langle\langle X_3\rangle\rangle \cdots X_{n-1}\langle\langle X_n\rangle\rangle .$$

- Two approaches for validating certificates:
  1. X.509v3 (Directory Authentication Framework).
  2. OpenPGP.
- Certificate signature chains are more flexible in OpenPGP than in X.509v3:
  - A key may have several signatures.
  - Signatures may have different levels of trust.
  - A Version 4 key is signed by its owner.

Key Storage

- The storage of cryptographic keys is problematic.
- The integrity of public keys must be protected, while both the integrity and the confidentiality of secret and private keys must be protected.
- Keys cannot be safely stored on multi-user computer systems, even as an encrypted file.
  - The key used to encrypt the file will reside in memory at some time.
  - Keystrokes can be captured.
- Keys need to be stored on a special dedicated device such as a smart card.

Key Revocation

- A key revocation makes a key invalid before it is set to expire.
- Key revocation requires:
  1. Authorization to issue the revocation.
  2. Timeliness in the communication of the revocation.
- A certificate revocation list is a list of certificates that are no longer valid.
  - Under X.509v3, the list is signed by the issuer of the certificates.
  - Under OpenPGP, the list is signed by the issuer or owner of the certificates.

Summary

- There are two principal kinds of encryption:
  1. Conventional encryption used for confidentiality.
  2. Public key encryption used for integrity, digital signature, and nonrepudiation.
- Key management is perhaps the most challenging aspect of cryptography.
  - Secure session key distribution is difficult.
  - Binding identity to public keys is problematic.
  - It is tricky to devise fully secure communication protocols.