Communications Security Principles

This lecture describes the principles used to ensure the secrecy,integrity, and authenticity of communications.

Introduction

Data communication security has the following goals:

secrecy ensures that the contents and sometimes the disclosure of information about the communication ("metadata") are not disclosed

authentication identifies the originator

integrity ensures that the message has not been modified by the communication system

Exercise 1: Which security goal(s) might need to be met by each of the following: signing into D2L, downloading a computer program, checking that a document is not forged?

Security can be applied at various layers:

- at the link layer. A typical example is wireless LAN encryption that protects against eavesdropping by other in range,
- at the network layer. A typical example is a virtual private network (VPN) which provides an encrypted "tunnel" between two remote sites.
- at the application layer, or "end-to-end". A typical example is secure HTTP or HTTPS which provides security between browser and a server applications.

The choice between these will depend on the nature of the threat, resources and policies.

Data communication methods covered below cannot protect from various other threats such as compromised hardware or software or human error or wrongdoing. These factors are often more significant threats than technical flaws.

For various reasons it's usually impossible to prove, in a mathematical sense, that a communication system is secure. Because the information being communicated is often valuable, there is a constant incentive to find ways to subvert communication security. Over time, all security systems have been "broken" and there is no reason to expect this trend will end.

Since so much effort has been devoted to subverting security mechanisms, it is important to choose well-tested security methods and to follow "best practices." Ad-hoc solutions are seldom effective.

Security mechanisms always have a cost. It may be in the form of lower reliability, more expensive hardware, more processing time, or inconvenience for users. Selection of appropriate security mechanisms therefore requires a cost-vs-benefit analysis to ensure the cost is appropriate for the value of the information being protected. Unfortunately, the benefits of communication security are often hard to quantify, particularly with new systems or where the threats are not well known.

Exercise 2: What factors might you take into account in evaluating the benefits of communication security for: an on-line computer game, a SCADA (supervisory control and data acquisition) link controlling a flood-control system, on-line backup of student records?

Although they can be very important, topics of computer security such as software integrity or user authentication are beyond the scope of this course.

Encryption

The basic tool for communication security is encryption. Modern encryption techniques use a publicly-known algorithm to convert "plaintext" into "ciphertext" using a secret "key".

If implemented correctly, the ciphertext cannot be decrypted without knowing the key.

We always assume the encryption algorithm is known to everyone and only the key is secret. The use of secret (and thus untested) algorithms is known as "security by obscurity" and is poor practice.

Cryptanalysis is the process of trying to deduce the key and/or plaintext from the ciphertext. Since in many situations the plaintext will be known, encryption algorithms are designed to prevent cryptanalysis with "known-plaintext" attacks.

The simplest (and the often only known) cryptanalytic attack is to try all the possible keys until a recognizable message is decrypted. For these types

of attacks the difficulty of decrypting the message increases exponentially with the length of the key.

Exercise 3: If you could test one key per nanosecond, how long would it take, on average, to find the key if it was 32 bits long? 128 bits? 1024 bits? Note that the age of the universe is approximately 450×10^{15} seconds.

In practice, security systems are most often subverted by the flaws in the software (e.g. bugs or malware), hardware (information leakage) or people (disgruntled or dishonest employees).

Symmetric Encryption

In symmetric encryption the same key is used for encryption and decryption. This requires some way to securely distribute keys.

There is only one type of encryption that can be proven to be unbreakable – a "one-time pad." In this case the key is a truly random sequence of bits as long as the plaintext. The key is then exclusive-or'ed with the plaintext.

Typically a key is used to periodically distribute a separate "session" key which is then used to encrypt the plaintext.

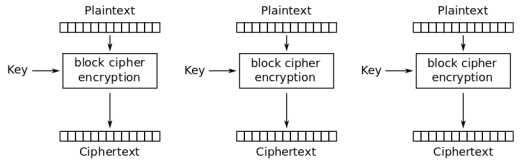
To avoid plain-text attacks the encryption algorithm must make it difficult to reproduce the key if given the ciphertext and plaintext.

More common symmetric encryption techniques are block ciphers such as DES (Data Encryption Standard, adopted in 1977 and now obsolete) and AES (Advanced Encryption Standard, adopted in 2001 and currently the most commonly used symmetric encryption algorithm).

AES encrypts data in blocks of 128 bits under control of keys of 128 or 256 bits. The algorithm was designed for efficient implementation in software. It consists of multiple "rounds" that rearrange (permute) and re-map the values of the 16 bytes in each block.

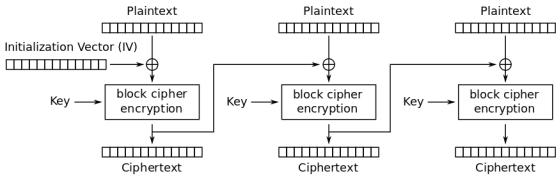
A symmetric encryption algorithm is a function with two inputs (plain text block and key) and one output (ciphertext). The algorithm can be used in different cipher "modes" to make certain types of cryptanalysis more difficult and suit different applications. Common cipher modes include[1]:

electronic codebook mode (ECB) The simplest approach. The key is used to encrypt the data. The drawback is that there is a 1:1 mapping between plaintext and ciphertext which makes cryptanalysis easier.



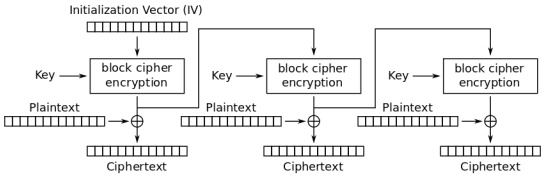
Electronic Codebook (ECB) mode encryption

cipher block chaining (CBC) To avoid this problem, the plaintext is first xor'ed with the previous ciphertext before being encrypted. An "initialization vector" (IV, transmitted in plain text) is used with the first plaintext block.



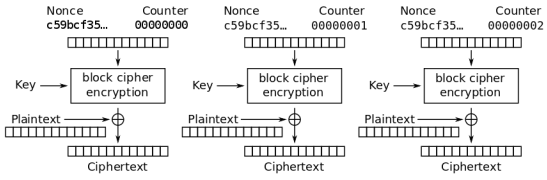
Cipher Block Chaining (CBC) mode encryption

stream cipher mode Also known as output feedback (OFB). This technique generates a sequence of bits which are xor'ed with the plaintext. The bits are generated by recursively encrypting the previous block of bits. The process begins with an IV.



Output Feedback (OFB) mode encryption

counter mode (CTR) An IV plus a counter (block index) value are encrypted and the result is xor'ed with the data. This allows starting the encryption or decryption at any point in the stream.



Counter (CTR) mode encryption

[1] Diagrams taken from Wikipedia

Exercise 4: Which mode(s) allow you to decrypt a portion of a file without having to decrypt everything before it? Which mode(s) allow you to continue decryption if the ciphertext contains errors?

Public Key Encryption

Public key encryption uses different keys for encryption and decryption. One key, the public key, is assumed to be known to everyone. The second key, the private key must be kept secret.

Public key algorithms simplify the problem of key distribution and for this reason have been widely adopted. However, the encryption and decryption algorithms are much slower than symmetric ones. PKE algorithms are typically used only to transfer a session key which is then used for symmetric encryption.

Public key algorithms are designed so that the public and private keys cannot be (easily) derived from each other. Several public key algorithms have been developed, the most popular of which is known as RSA (after the names of the inventors).

Two devices that want to communicate securely first exchange their public keys and then use these public keys to securely transfer a random session key.

Hash Algorithms

It is often useful to be able to compute a value that is similar to a CRC in that it is a function of the contents of a document but has the additional property that it cannot be easily reversed – it should be very difficult to create a document that matches a particular checksum (this is not the case with a CRC). These values are called cryptographic hashes or message digests and can be used to verify that the document has not been altered.

The most common hash algorithm is SHA-1 (Secure Hash Algorithm 1). It produces a 160-bit hash value from a document.

Authentication

Authentication means verifying identity. For example, authenticating the author of an e-mail or the company responsible for a web site.

In addition to exchanging session keys, public key algorithms are widely used for authentication. Some public key algorithms (such as RSA) have the property that the encryption and decryption can be done in reverse order and still produce plaintext. A document (or, more typically, a hash of that document) can be "signed" by first "decrypting" it with a private key. A recipient can use the public key to encrypt the signature and check to see if the value matches the document hash. If so, then the document must have been signed by someone in possession of the secret key corresponding to the private key. The document can be an arbitrary file, including software.

Public Key Infrastructure

The use of public keys for authentication requires a method to authenticate the public keys themselves. This is done by using certificates. These are documents that contain indentifying information along with the public key. The certificates are themselves signed by a Certificate Authority (CA) so that the certificate can be verified.

A CA also publishes a certificate with its public key so the certificates that it signed can be authenticated. One CA can also sign another CA's certificate. Ultimately this "chain of trust" must lead to a certificate that cannot be authenticated by means of its signature. These certificates are typically supplied by a trusted party such as a software manufacturer who has verified the certificates in some other way. For example, a browser comes configured with several dozen CA certificates, all of which are implicitly trusted to authenticate other certificates.

The system of signing certificates that establish this chain of trust between CAs is called a public key infrastructure (PKI).

X.509 is the ISO-standard certificate format that has been widely adopted.

Certificates can have restrictions such as being valid only for specific purposes (e.g. only for authenticating the author of an e-mail or only for signing other certificates), or being valid only for certain dates.

If a user's private key has been disclosed and it's not possible to wait until the certificate expires, it is possible to revoke a certificate. To do this it is necessary to publicize that the certificate is no longer valid so that those who obtained the private key cannot impersonate the original owner. A CA can publish a certificate revocation list (CRL) with the certificates that should not be trusted.