CERTIFICATION MAGAZINE

Cryptography Techniques for Secure Communications

PRINT EMAIL SHARE Back**By Robert Winding — *July 2005***

Chris is Motivated.
Chris gets recognition.
Chris has an ISACA certification.
www.isaca.org/certification
Exam date: 11 June 2011
Registration deadline: 6 April 2011

Electronic communication is the lifeblood of many organizations. Much of the information communicated on a daily basis must be kept confidential. Information such as financial reports, employee data and medical records needs to be communicated in a way that ensures confidentiality and integrity. This makes good business sense and may even be regulated by legislation like the Health Insurance Portability and Accountability Act (HIPAA). The problem of unsecure communication is compounded by the fact that much of this information is sent over the public Internet and may be processed by third parties, as in e-mail or instant messaging (IM).

Cryptography Basics

Cryptography can be used to provide message confidentiality and integrity and sender verification. The basic functions of cryptography are encryption, decryption and cryptographic hashing. In order to encrypt and decrypt messages, the sender and recipient need to share a secret. Typically this is a key, like a password, that is used by the cryptographic algorithm. The key is used by the sender to encrypt the message (transform it into cipher text) and by the recipient to decrypt the message (reverse the cipher text back to clear text). This process can be done on a fixed message, such as an e-mail, or a communications stream, such as a TCP/IP connection.

Cryptographic hashing is the process of generating a fixed-length string from a message of arbitrary length. If the sender provides a cryptographic hash with the message, the recipient can verify its integrity. Modern cryptographic systems are based on complex mathematical relationships and processes. Let�s focus on the common cryptography standards used to secure computer communications and how they are used.

The three basic types of cryptography in common use are symmetric key, asymmetric (public) key systems and cryptographic hash functions. Typically, the strength of a crypto system is directly related to the length of the key. This assumes that there is no inherent weakness in the algorithm and that the keys are chosen in a way that fully utilizes the key space (the number of possible keys). There are many kinds of attacks that can be used against crypto systems, but these are beyond our scope here. That said, if you use public algorithms with no known vulnerabilities, use reasonable key lengths (most defaults are fine) and choose good keys (which are normally chosen for you), your communications will be very secure.

Symmetric key cryptography uses the same key to encrypt and decrypt data. Some common symmetric key algorithms are the Data Encryption Standard (DES), Triple DES, Blowfish and the Advanced Encryption Standard (AES). DES is ineffective because it uses a 64-bit key and has been broken. Be careful, because some crypto security, like Microsoft�s Windows XP Encrypted File System (EFS), defaults to DES and must be changed to provide good security.

The main advantage of symmetric key cryptography is speed. The principle problems with this system are key distribution and scalability. Keys need to be distributed securely, and each secure channel needs a separate key. Symmetric key systems provide confidentiality but do not provide authenticity of the message, and the sender can deny having sent the message.

Asymmetric (public) key cryptography uses a pair of mathematically related keys. Each key can be used to encrypt or decrypt. However, a key can only decrypt a message that has been encrypted by the related key. The key pair is called the public/private key pair. Some common public key systems are Rivest-Shamir-Adelman (RSA), Diffe-Hellman and Digital Signature Standard (DSS).

Asymmetric key systems solve the key distribution and scalability problems associated with symmetric systems, although it�s not trivial to manage and implement a public key infrastructure (PKI). However, you can take advantage of companies like Thawte, Verisign and PGP Corp. that provide key distribution and trust services. Asymmetric key systems provide a greater range of security services than symmetric systems. They provide for confidentiality, authenticity and nonrepudiation. The principle problem with these systems is speed. It takes significantly more computer resources to encrypt and decrypt with asymmetric systems than symmetric ones.

Cryptographic hash functions take a message of arbitrary length and compute a fixed signature, often called a message digest, for the message. This can be done for a file, e-mail message or your entire hard-drive image. The main properties of these functions are that it is difficult to find different files that produce the same digest and that the function is one-way. Therefore, it is not computationally feasible to recover a message given its digest.

Two common examples of hash functions are the Secure Hash Algorithm (SHA), commonly SHA-1, and Message-Digest algorithm 5 (MD5). SHA-1 is used in many common security applications including SSL, TLS, S/MIME and IPSec. MD5 is generally used to create a digital fingerprint for verifying file integrity.

So symmetric is fast, but exchanging keys is a problem; and asymmetric has more security services, but it�s slow. The solution: Combine them in a hybrid system. This is what is done in the digital-certificate-based crypto systems that are in common use for e-mail, IM and SSL Web traffic. The basic idea is to use an asymmetric system, as is done with Diffie-Hellman key exchange, to exchange a symmetric key to do the bulk of the data encryption.

CAs and Webs of Trust

Trust is one of the services provided by certificate vendors. There are two common trust models: certificate authorities (CA) and webs of trust (WOT).

The CA model is based on the notion that you trust the CA to vouch for the validity and integrity of the certificate that is being presented to you. For example, you might visit a Web site to purchase something, and the site will use SSL to secure the site. As part of that process, your browser asks the server to authenticate its identity. The server does this by providing its digital certificate, typically signed by the CA. By signing the certificate, the CA is assuring you that it has verified the identity of the entity that it gave the certificate to. This might involve verifying business records, domain registration information, company contacts, etc. Your Web browser has many CA certificates in its Trusted Root Library. It uses the appropriate root certificate to verify that the certificate presented by the Web server is in fact signed by the CA. Other parameters, such as expiration date, also are checked. Without using the CA model, you would be told who signed the certificate (it might be self-signed), and then you would have to independently decide if you trusted the site.

WOTs use a process of elevating trust in a certificate by having others in the web vouch for the owner of the certificate. This usually starts as simply verifying an e-mail address by requiring you to respond to an e-mail before you can obtain a certificate that uses that e-mail address for identity. In Thawte�s WOT, you must physically meet a certificate notary and present various forms of identification to elevate your level of trust. The notary will then add trust points to your certificate, allowing people to have a greater certainty that you are who you claim to be. Self-signing certificates is fine if you only communicate with people you know, and they know how to verify that the certificate is yours.

When you communicate with someone who doesn�t know you, then a CA or WOT enables them to have confidence that it is really you. A certificate also can be revoked, by the CA or the owner, so that people will know that the certificate is no longer valid. Certificate revocation can be a more complicated problem because it assumes that everyone will check for revocation each time a certificate is used.

Viewed 5300 times.

PRIVACY POLICY