CRYPTOGRAPHY

Cryptography is a physical process that scrambles information by rearrangement and substitution of content, making it unreadable to anyone except the person capable of unscrambling it. With the shear volume of sensitive Internet transactions that occur daily, the benefit of securing information using cryptographic processes becomes a major goal for many organizations. Since no cryptographic system is foolproof, the idea is to make the cost of acquiring the altered data greater than the potential value gained [3:27]. Essentially, it becomes an issue of deterrence.

Generally, all cryptographic processes have four basic parts:

> **Plaintext -** Unscrambled information to be transmitted. It could be a simple text document, a credit card number, a password, a bank account number, or sensitive information such as payroll data, personnel information, or a secret formula being transmitted between organizations.
>
> **Ciphertext** - Represents plain text rendered unintelligible by the application of a mathematical algorithm. Ciphertext is the encrypted plain text that is transmitted to the receiver.
>
> **Cryptographic Algorithm - A** mathematical formula used to scramble the plain text to yield ciphertext. Converting plain text to ciphertext using the cryptographic algorithm is called encryption, and converting ciphertext back to plain text using the same cryptographic algorithm is called decryption.
>
> **Key** - A mathematical value, formula, or process that determines how a plaintext message is encrypted or decrypted. The key is the only way to decipher the scrambled information.

HISTORY

Cryptography has a long history, actually dating back to the time of Julius Caesar who encrypted messages by substituting each letter in the document by the letter that appears three positions further down the alphabet. Both world wars provided some scientific advances to cryptographic processes but it wasn't until the advent of the radio that the need for cryptography became apparent.

In 1967, the appearance of David Kahn's best selling book entitled *The Codebreakers* introduced the general public to the secret world of cryptography. Interest continued to develop throughout the 1970s and 1980s much to the chagrin of the National Security Agency (NSA) which became America's official cryptographic organ. With secrecy as its goal, the NSA tried repeatedly, through coercion and intimidation, to prevent the widespread public dissemination of information about cryptography. However, with improved communication technology, the growing need for access control, electronic payments, and corporate security and the advent of the Internet, cryptography finally became public and world wide.

Prior to the Internet, use of cryptography was structured around symmetric cryptography (often called conventional cryptography) and frequently involved the use of special-purpose hardware to execute the algorithms. Symmetric cryptography uses the same key to both encrypt and decrypt information. That implies that both sender and receiver must possess the same key. Prior to the Internet, this approach proved satisfactory and was the accepted approach to cryptography. Symmetric cryptography provided a great advantage over normal forms of communication but posed some new problems. For instance, as the use of cryptography expanded into the public domain, where organizational networks often spanned multiple countries with thousands of employees in each, safeguarding the transportation and safe keeping of thousands of secret keys proved a daunting task. The risk of key exposure rises dramatically as the number of users increases making symmetric cryptography more trouble than it's worth for large organizations.

With the advent of the Internet, the use of symmetric cryptography proved to be an even greater liability because sender and receiver often never met or even knew each other. As is often the case, revolutionary technology such as the Internet often forces expansion of existing technologies. Public key cryptography, introduced in 1976 by Whitfield Diffie and Martin Hellman, was developed to accommodate the tremendous risks inherent in any Internet transmission, risks that symmetric cryptography couldn't overcome. Unlike symmetric cryptography, public key cryptography uses two keys, one public and one private. The private key never has to leave the owner, negating the high risk of transporting keys to each document recipient. The public key can be made public so everyone can have access to it by simply downloading it from the Internet. The risks of safeguarding a highly secret key during distribution to users disappears, making public key cryptography ideally suited for the Internet, large distributed systems, and big corporate networks. An example of how public key cryptography actually works will be given below under examples.

ALGORITHM TYPES AND STRENGTHS

Let's take a closer look at both symmetric and public key cryptography. As a subset of cryptography, cryptographic algorithms can be divided into two categories:

> **Stream algorithms** – Operate on plaintext one byte at a time, where a byte is a character, number, or special character. The process is inefficient and slow.

Block algorithms – Operate on plaintext in groups of bytes, called blocks (hence the name block algorithms or block ciphers). Typical block sizes for modern algorithms is 64 bytes, small enough to work with but large enough to deter code breakers. Unfortunately, with the current speed of microprocessors, breaking a 64-byte algorithm using brute force is proving to be to relatively easy task [4].

Encrypted information is only as good as the key required to decrypt it. In theory, any key can eventually be obtained and the encrypted information decrypted successfully. It's really a question of the time required to break the key. In the early years of cryptography, before the computer, even small keys were nearly impossible to break. The advantage gained with computers, however, is pure speed.

Modern computers can operate at incredible speeds, trying literally billions of permutations each second. Generally, using brute force only (trying every possible permutation in a linear fashion), the time required to reveal a symmetric key increases exponentially with the length of the key. For example, a 32-byte key would take about 2^{32} steps to solve. Solving a 32-byte key could be achieved on your home computer in a short time. A 40-byte key would take about 2^{40} steps to break. Again, you could do it on your home computer in about a week's time. Beyond 40 bytes, processing time begins to rise dramatically. For example, a 56-byte key would take a large number of personal computers working simultaneously in a distributed approach to solve in even a few months time. Larger keys such as 64 bytes can be broken by large organizations such as governments or criminal organizations who have access to very powerful computers such as supercomputers. The process, however, would still take several years. Larger keys, with 80 to 128 bytes remain relatively safe well into the future [5].

Public key algorithms can use much larger keys to achieve secrecy. Where breaking symmetric keys is usually done through brute force, public key algorithms involve deriving the matching secret key from the public key. The process depends on the kind of encryption adopted but generally involves the use of prime factors. 768-byte keys are safe for the near future and 1028-byte keys are safe for the foreseeable future, whereas 256-byte keys can be easily cracked by personal computers [5:1].

ADVANTAGES AND DISADVANTAGES

Even though public key cryptography is the accepted standard, it's not foolproof. For this reason, it has not completely replaced symmetric cryptography. Here are some of the main advantages and disadvantages [4].

Advantages:

1. The biggest advantage of public key cryptography is the secure nature of the private key. In fact, it never needs to be transmitted or revealed to anyone.

2. It enables the use of digital certificates and digital timestamps, which is a very secure technique of signature authorization. We will look at digital timestamps and digital signatures in a moment.

Disadvantages:

1. Transmission time for documents encrypted using public key cryptography are significantly slower then symmetric cryptography. In fact, transmission of very large documents is prohibitive.
2. The key sizes must be significantly larger than symmetric cryptography to achieve the same level of protection.
3. Public key cryptography is susceptible to impersonation attacks.

EXAMPLES

The ABC company maintains payroll information for a variety of organizations. This payroll information is frequently transmitted over the Internet from participating companies. For security reasons, the ABC company conducts all of its Internet transactions using public key cryptography. The company owns both a public and a private encryption key. The public key is made available to all participating organizations and in fact is openly available to anyone who wants to download it from the ABC website. The private key is kept secure in a bank vault at ABC headquarters. When the XYZ company wants to transmit its payroll data to the ABC company, it first encrypts the data using the ABC company's public key. Once it's encrypted, the scrambled payroll data is transmitted securely over the Internet to the ABC company's processing department. If the information is intercepted along the way, all the interceptors will see is scrambled information. Even if they have the public key, which is very possible, they will not be able to unscramble the information. Only the private key can do that. Once the information is received by ABC, the private key is used to unscramble the information, allowing the processing department to process the payroll.

Using symmetric cryptography the ABC company would have to deliver, through some secure means (such as a courier), a copy of its one and only private key. Since the same key is used to both encrypt and decrypt the information, both sender and receiver must have a copy. So if XYZ is a new client for ABC, ABC must send XYZ a copy of the secret key so that XYZ can then encrypt its payroll information and transmit it to ABC. ABC, using the same key, decrypts XYZ's information and processes the payroll data. Since a system is only as strong as its weakest link, key security during transmission becomes as important for XYZ as encrypting the data.

As mentioned earlier, public key cryptography lends itself to a new technology called digital signatures. Digital signatures involve a reversing of the normal public/private encryption/decryption process. Here is an example that demonstrates its use. Suppose Mary wants to send the ABC company a request for a special document. Before the ABC company can send that document, they must be assured that the requestor is actually

Mary. A digital signature can verify Mary’s validity to ABC in the following way. Mary first encrypts her name using her private key. She then encrypts the request along with the encrypted name using the ABC company’s well-known public key. When the ABC company receives the message, it decrypts the request using its private key and then decrypts the signature using Mary’s well-publicized public key. If the name decrypts successfully, then it must be Mary’s signature since she is the only one who could have encrypted it with her secret private key. The request can be safely processed.

Digital signatures are gaining popularity in many Internet transactions involving signature verification such as contracts and other legal negotiations as well as court documents. Recent enhancements to digital signatures include digital timestamps. Digital timestamps apply a “when” criteria to a digital signature by attaching a widely publicized summary number to the signature. That summary number is only produced at some given point in time, essentially linking that signature to a certain date/time. It’s an especially effective technology since it doesn’t rely on the security of keys.

I mentioned earlier that for large documents, use of public key cryptography is prohibitive because transmission speeds are so slow. By using something called a digital envelope, the best of both symmetric (transmission speed) and public key (security) cryptography can be used. Here is an example of how a digital envelope works. Mary wants to send a very large document to her main office overseas. Because of its sensitivity, Mary believes it should be sent using public key cryptography but knows she can’t because it’s too large. She decides to use a digital envelope.

Mary first creates a special session key and uses this key to symmetrically encrypt her document. That is, she uses a symmetric cryptographic algorithm. She then encrypts the session key with her organization’s public key. So now the document is encrypted using symmetric cryptography and the key that encrypted it is encrypted using public key cryptography. The encrypted key is called the digital envelope. She then transmits both the key and the document to the main office.

At the main office, the company’s private key is used to decrypt the session key. Then the session key is used to decrypt the document. Transmission was fast and just as secure as using public key cryptography exclusively [1:24]. Digital envelops offer the benefits of both approaches without sacrificing security.

FUTURE DEVELOPMENTS

There are two areas that show promise in the field of cryptography: Quantum cryptography and DNA cryptography. Quantum cryptography attempts to achieve the same security of information as other forms of cryptography but through the use of photons, or packets of light. The process, though still in experimental stages, makes use of the polarization nature of light and is proving to be a very promising defense against eavesdropping [4].

DNA cryptography makes use of specially selected DNA strands whose combination results to a specific solution to a problem. Still in its infancy, DNA cryptography looks promising [4].

REFERENCES

1. Stein, Lincoln D., Web Security, New York: New York Addison-Wesley, 1988

2. McGraw, Gary, Felten, Edward F, Securing Java, New York: New York, John Wiley & Sons, 1999

3. Feghhi, Jalal, Feghhi, Jalil, Williams, Peter, Digital Certificates, Addison Wesley Longman, Inc., 1999

4. Christoyannis, Costas, http://www.hack.gr/users/dij/crypto/

5. SSH Communications Security, http://www.ssh.fi/tech/crypto/intro.html#algorithms