Symmetric vs. Asymmetric Encryption

.Introduction

This paper discusses some of the basic considerations of asymmetric and symmetric encryption from the angle of the ultimate security that is provided.

Examples of Symmetric encryption include: Data Encryption Standard (DES), Triple DES (3DES) and Advanced Encryption Standard (AES).

Asymmetric encryption is commonly referred-to as Public-Private Key encryption or just Public Key encryption. Perhaps the best known example is Pretty Good Privacy (PGP).

.General Description of Symmetric and Asymmetric Encryption

..How Symmetric Encryption Works

Symmetric encryption involves the use of a key that we will call the Symmetric Key. In today's computer-based systems this Symmetric Key is a series of numbers and letters. Example:

f8kW2B60mVa2Kjue

This Symmetric Key will be used to encrypt a message. This very same Symmetric Key must be used to decrypt the message. There is only one key in Symmetric Encryption - the Symmetric Key.

...Example

Alice wants to send an encrypted message to Tom, say over the Internet. Therefore, Alice has to find a way to safely get the Symmetric Key to Tom.

Inasmuch as Alice encrypts her message before sending it over the Internet, it is clear that she does not consider the Internet to be a secure way to send messages. She fears, rightly so, that her message could somehow get into the hands of some person other than the intended recipient. Since Alice's premise is that the Internet is not secure, she may use traditional postal mail to send the Symmetric Key to Tom. She could also hand the Symmetric Key to Tom, saying, "Any time in the future that I want to send you a secure message, I will use this Symmetric Key. With this Symmetric Key you can decrypt the message I send to you."

Interestingly, this one Symmetric Key can be used for either of these two parties to send encrypted messages to each other. As noted later in this paper, Asymmetric Encryption does not work this way.

..How Does Asymmetric Encryption Work?

Asymmetric Encryption uses two keys, a Private Key and a Public Key.

- The Public Key is used to encrpyt a message.
- The Private Key is used to decrypt a message.

This is where the term "asymmetric" comes from. Different keys are used on each end of an encrypted communication between two parties.

Someone who wants to receive an encrypted message will use PC-based software to generate these two keys, as a pair of keys. Then the Public Key will be sent to the party who has the message to send. When the encrypted message is received, the Private Key will be used to decrypt it.

There is an important feature here. Assume that the Public Key is sent over the Internet In many cases we don't care if the Public Key is intercepted. The Public Key is only used to encrypt a message. In many cases it does not matter if a hostile party has the Public Key. All they would gain from that is the ability to send their own encrypted message. As noted later in this paper, this could be a problem of a different nature.

...Example

Alice wants to send an encrypted message to Tom, say over the Internet. In order to do this, Alice has to request that Tom create a Public Key - Private Key pair, and send the Public Key to her. Tom does not need to find a way to safely get a key to Alice because Tom will send the Public Key and that does not allow an intercepting party to decrypt the message that he sends to Alice.

Note that the Public Key that Tom sends to Alice is for transmittal of **messages from Alice to Tom**. For the reverse, **messages from Tom to Alice**, Alice must create her own Public Key-Private Key pair and send that Public Key to Tom.

.Strengths and Weaknesses of Symmetric and Asymmetric Encryption

..What Is Good About Symmetric Encryption?

Symmetric Encryption has almost always been the approach-of-choice for governments. Since the financial resources of a government to evolve cryptography (or any initiative for that matter) are considerable, thus far the "big money" that has "big secrets to protect" is going with Symmetric Encryption.

The popular Symmetric Encryption approaches have enjoyed speed advantages over Asymmetric approaches. In many applications, this speed of encrypting and decrypting is quite important.

..What Is Undesirable About Symmetric Encryption?

The principle disadvantage of Symmetric Encryption involves a logistics problem of conveying the Symmetric Key. Recall that in Symmetric Encryption, there is only one key, the Symmetric Key. An example will illustrate.

...Example 1

Perhaps the most common occurrence of a situation in which encryption is needed and Symmetric Encryption does not currently serve well is that of a Web-based secure form.

In this situation, the User (a person) operates a PC, using a Web browser such as Internet Explorer or Navigator, to connect to an Internet Web server that will take an order for some merchandise. A "secure form" is the computer-based form that the User fills in by typing into his/her browser. The User wishes to enter his/her credit card number into this secure form and have the computer/browser send this information, over the Internet, to the Web server taking the order. The User wants this transmission of the credit card

number to be encrypted so that if the transmission is intercepted, it will appear as meaningless gibberish to the person who intercepted it.

If Symmetric Encryption were to be used here, then somehow, ahead of time, this User must have communicated with the company that runs the web server and conveyed to them the Symmetric Key needed for Symmetric Encryption. This is the only way the credit card information could be decrypted by the company that is taking the order.

This can be quite inconvenient to both the User and the merchant and would tend to stifle internet commerce (e-commerce).

..What Is Good About Asymmetric Encryption?

With Asymmetric Encryption, the Public Key can be sent over the Internet. If it is intercepted it only allows the intercepting party to create his/her own encrypted message. It does not allow the intercepting party to decrypt someone else's message.

Let's see how this works in the aforementioned case of a secure form over the Internet.

...Example 1

As mentioned earlier, the User (a person) operates a PC, using a Web browser such as Internet Explorer or Navigator, to connect to an Internet Web server that will take an order for some merchandise. A secure form is the computer-based form that a User fills in by typing into his/her browser. The User wishes to enter his/her credit card number into this secure form and have the computer/browser send this information to the Web server taking the order. The User wants this transmission of the credit card number to be encrypted so that if the transmission is intercepted, it will appear as meaningless gibberish to the person who intercepted it.

In the Asymmetric Encryption approach, the User can request a Public Key from the Web server that is taking the order. The Web server can create a Public Key - Private Key pair at once, and send the Public Key to the User. The User can then encrypt the secure order form that contains his/her credit card information and send it safely over the Internet to the Web server. The Web server can then use its Private Key to decrypt the secure form and thereby complete the order for merchandise.

...Example 2

Another common use of Asymmetric Encryption is the simple sending of confidential messages to some party. This example has been given earlier in this paper.

..What Is Undesirable About Asymmetric Encryption?

...Strength of Encryption

It is difficult to compare strength of encryption of different approaches to encryption unless the application for which encryption is required, the implementing hardware and its constraints as well as User procedures are closely controlled.

As an example of this, consider comparing the required key length, in bits, between a Symmetric Encryption and an Asymmetric Encryption. The following chart is from the NIST publication "Key Management Guideline", DRAFT, dated Nov 1-2, 2001.

RSA encryption, used in Asymmetric Encryption such as PGP, is noted.

Equivalent Strengths Table

Enc. Bits	Symmetric Alg.	RSA
112	3DES	k = 2048
128	AES-128	k = 3072
192	AES-192	k = 7680
256	AES-256	k = 15360

When reviewing the above table, a reader might be led to believe that, for example, an RSA-based encrypted message using 15,360 bits is just as secure as an AES encrypted message with 256 bits. AES, however, provides superior security, and depending upon many other factors, the AES may be in fact be vastly better.

....Why Symmetric May Be Vastly Better 1 - CPU Speed, Memory, Power

The CPU time needed for the RSA encryption is more than that required for AES. AES was carefully tuned to allow very low power chips, embedded in smart cards, to encrypt and decrypt. AES was also tuned to be efficient with required memory.

If AES at 256 bits key length takes fewer CPU cycles and less memory to encrypt/decrypt than RSA at 256 bits, consider comparing AES at 256 bits to RSA at 15360 bits.

....Why Symmetric May Be Vastly Better 2 - Back Doors / Key Recovery

Various online and print sources indicate that one of the most popular Asymmetric encryption systems has key recovery built in. Who has the access to this key recovery? Are you concerned about that entity having instant access to your messages? Also, assuming that key recovery is indeed buried in that product, are you concerned that a math graduate student hacks on one of his/her own encrypted messages and figures out the key recovery process and publishes the results online?

....Why Symmetric May Be Vastly Better 3 - Advances in Factorization

Central to the Public Key - Private Key approach is the multiplication of two large prime numbers. Central to the cracking (unauthorized decrypting) of a message encrypted with this scheme is factorization to find the original two numbers. Advances in the mathematical techniques of factorization continue, as do advances in CPUs, including those dedicated to cracking, continues.

The recent records for this factorization/cracking are:

1999: An RSA-512 took 2 months on 300 PC's, each with 64 Mbytes of memory.

2001: Mathematician Daniel Bernstein published "Circuits for Integer Factorization", which roiled the industry in debate and threatened the confidence in the widely used RSA Public Key - Private Key encryption system. Bernstein used special-purpose chips in a configuration of parallel processing to reduce the cost of factoring. In one writing he is quoted as positing the vulnerability of 1536 bit key lengths.

2003: On December 3rd, a team at the German Bundesamt für Sicherheit in der Informationstechnik (Federal Bureau for Security in Information Technology; BIS) announced the factorization of the 174-digit number:

1881 9881292060 7963838697 2394616504 3980716356 3379417382 7007633564 2298885971 5234665485 3190606065 0474304531 7388011303 3967161996 9232120573 4031879550 6569962213 0516875930 7650257059

known as RSA-576 (see http://mathworld.wolfram.com/news/2003-12-05/rsa/).

Various commercial consultant organizations offer opinions on Bernstein's work and other cracking efforts and offer advice ranging from: *RSA-1024 bits is good for 20 years*, to *RSA-2048 bits is needed now*.

Note that when computing cracking times and costs, almost all papers describing the necessary hardware capability and architectures center around devices and systems that are commercially available **now**.

Inasmuch as Bernstein described circuitry for fast factorization, it is entirely possible that an organization with sufficiently deep pockets can build a large scale version of his circuits and effectively crack an RSA-1024 bit message in a relatively short period of time, which could range anywhere from a number of minutes to some months.

A large majority of deployed systems, such as HTTPS, SSH, IPSec, S/MIME and PGP, utilize RSA as the public key algorithm, and these systems typically do not use keys up to 1,024-bits. In addition, vendors supplying the underlying software for these protocols most likely do not offer support for any more than 1024 bits. An entity capable of breaking all of the above will have access to virtually **any** corporate or private communications and services that are connected to the Internet.

....Why Symmetric May Be Vastly Better 4 - Authentication Required

It was mentioned earlier in this paper that sending a Public Key over the Internet did not pose problems in certain cases since it did not allow an intercepting party to decrypt messages. It does allow an intercepting party to create a message and send it as if from the intended party.

Example:

Alice wants to send an encrypted message to Tom, say over the Internet. In order to do this, Alice has to request that Tom create a Public Key - Private Key pair, and send the Public Key to her. Tom does so, but when he sends the Public Key to Alice, it is intercepted by Dan.

Now Dan can send an encrypted message to Tom, falsify the email return address to make it look like it is from Alice. When Tom receives the email and when he decrypts the message, he thinks it is from Alice.

This problem is addressed by the use of Authentication within the framework of a Public Key Infrastructure (PKI). PKI is beyond the scope of this white paper. Suffice it to say that PKI has its own set of problems, also beyond the scope of this paper. The point here is that Symmetric Encryption does not have this inherent Authentication weakness.

....Why Symmetric May Be Vastly Better 5 - Non-Repudiation Required

It was mentioned earlier in this paper that sending a Public Key over the Internet did not pose problems in certain cases since it did not allow an intercepting party to decrypt messages, but that it does allow an intercepting party to create a message and send it as if from the intended party.

Example:

Alice wants to send an encrypted message to Tom, say over the Internet. In order to do this, Alice has to request that Tom create a Public Key - Private Key pair, and send the Public Key to her. Tom does so, and Alice sends him a message.

When Tom receives her message he is very unhappy about it. It doesn't matter why, he is just unhappy about it. When he confronts Alice with his unhappiness, she denies having sent it and suggests that someone intercepted the transmission of the Public Key over the Internet and sent Tom a message as if it had been from her.

In other words, Asymmetric Encryption does not inherently allow non-repudiation. Alice can easily say that the message might have been sent by someone else.

This problem is addressed by the use of Non-Repudiation within the framework of a Public Key Infrastructure (PKI). PKI is beyond the scope of this white paper. PKI has its own set of problems, also beyond the scope of this paper. The point here is that Symmetric Encryption does not have this inherent Non-Repudiation weakness.

.Summary and Conclusions

This paper has reviewed issues pertaining to ultimate security strength of the Symmetric and Asymmetric encryption methods.

Clearly each system has its strong points:

- Symmetric is ultimately much stronger at smaller key lengths. AES, an implementation of the symmetric method, is considerably more efficient in the use of CPU cycles and CPU memory.

- Asymmetric addresses the important problem of passing keys over an insecure network. It is a solution to the secure-form problem, with the attendant risks mentioned above.

In conclusion, Symmetric Encryption is the winner by a considerable margin.