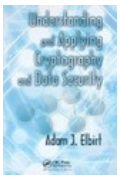
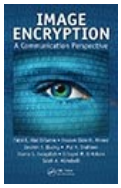
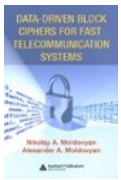
INFORMATION SYSTEMS SECURITY
The Final Word in Information Systems Security
AUERBACH

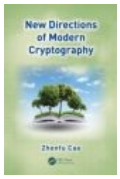


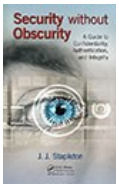


## New Books








# Introduction to Cryptography

**by Barry K. Shelton**

Tweet

The U.S. economy fundamentally changed in the last twenty years, as manufacturing and heavy industry moved overseas, replaced by a new focus on knowledge and data. This transformation has underscored the importance of safeguarding information through encryption. This article focuses on state-of-the-art encryption techniques used pervasively to protect data, such as personal identity, medical records, financial transactions, and electronic mail, to name a few.

A typical approach to security is to strike a balance between apparent risks to information and efforts to mitigate those risks. A common standard used to determine the level of security required is "commercial impracticability" - if it takes longer to access critical data than the timeframe within which its knowledge confers some benefit, practical security has been achieved. For example, if your credit card information is protected by a system that would take the most sophisticated hacker five years to unlock, but you obtain new credit card numbers every two years on average, there will be little benefit to 'breaking' the security scheme.

An important concept in security is that virtually any security system can and will be compromised eventually; it simply takes time. For example, the Japanese never broke the code employed with great success by the Navajo code talkers in the Pacific theatre during World War II, but their code was only employed for a few years.[1] The success of that code was the use of words in a foreign and little-known language to represent military messages. Had the Japanese efforts to decrypt the Navajo code focused more on linguistics than cryptography, it would have likely been just another broken security scheme in a long line of others.

**Encryption**

Although there are many ways to protect information from undesired access, including various physical security techniques that prevent any access from unintended receivers, it is most useful to safeguard data so that it can be transmitted over insecure networks, such as the Internet, without fear of compromise. Since the time of the ancient Egyptians, cryptography, or the art of secret writing, has been employed to keep key information private. History is replete with examples of the successes and failures of encryption; lives have been lost and the outcome of battles determined solely on the strength or weakness of a cipher.[2] This article examines the different forms of encryption, both symmetric and asymmetric, and evaluates the common algorithms and applications of encryption today. Although you won't be a cryptographer after reading this article, you will have a better appreciation of the pivotal role encryption plays in our lives today.

Encryption algorithms or ciphers are mathematical formulas or functions applied to data to transform the unprotected information, or *plaintext* or *cleartext,* into an unrecognizable format commonly referred to as *ciphertext*. There are generally two inputs to an encryption algorithm: a *key* and the plaintext itself.[3] In some cases the ciphertext is larger than its associated plaintext or the same size. The goal is to make the time it would take to recover or *decipher* the plaintext, having only the ciphertext and not the key, so long as to greatly exceed the time-value of the plaintext. Ideally, a strong algorithm and key combination should take at least millions of years to break, based on mathematical predictions. Naturally, if an interloper manages to somehow obtain the ciphertext and the key, deciphering the information is as straightforward as it is for the intended receiver, and therefore all security is lost.

Much of security is predicated on strong methods of keeping encryption keys sacrosanct, in order to force attackers to use *brute-force* methods, such as trying every possible key combination with the use of fast computers.[4] The ideal algorithm is strong, meaning that the algorithm itself is relatively impervious to direct attack, leaving attempts to derive or guess the key as the only practical avenue to breaking the encryption. The ideal encryption algorithm creates unique ciphertext from the same plaintext for each key permutation, among other traits.

So what exactly is a key? A key is simply a number with a predetermined length. Keys can be created or

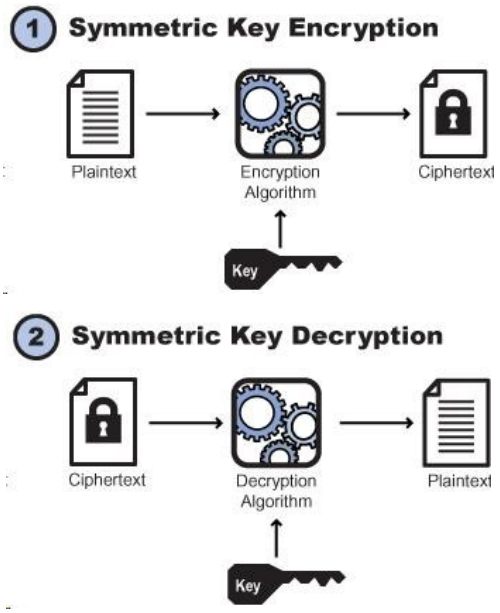
generated in many ways, but computers commonly generate them. Ideally, each key is truly random, meaning that any possible key combination is equally likely and that keys are not generated in a predictable fashion. A random number generator (RNG) or a pseudo-random number generator (PRNG) is frequently used for this purpose. The difference between an RNG and a PRNG is that the RNG autonomously generates random numbers, whereas a PRNG is computer-based and creates a somewhat random number based on *seed* values that are readily available within the computer.[5] A significant threat to any PRNG is the feasibility of regenerating the key if one can determine the seed values. Since true randomness only occurs in nature, research continues into how to generate completely random numbers based on detection of naturally occurring state information, such as electron spin and radiation decay. The key length or modulus determines how many combinations are possible, and is commonly expressed in bits. The number of key combinations is 2 raised to the power of the key modulus. For example, a 40-bit key has $2^{40}$, or over a trillion, combinations. While this sounds like an extraordinary number, it is an insecure key size, as all possible combinations can be tried within mere minutes by modern computers.

Encryption algorithms are divided into two families based on the key type: symmetric or secret key, and asymmetric or public key encryption. In symmetric key encryption both the sender (encrypter) and receiver (decrypter) use the same secret key, so named because the strength of the system relies on the key being known only to the sender and receiver. In asymmetric key encryption, the sender and receiver each have distinct but mathematically related keys.

**Secret Key Encryption**

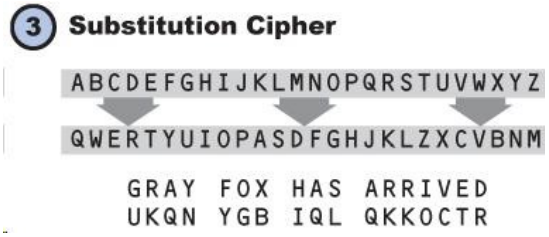
Symmetric encryption is the oldest form of encryption and has been used to safeguard communications for over three thousand years. All secret key algorithms or systems require that the party generating the key share or transfer it to the other party in a secure manner. If the key is not transferred by some means that prevents its interception by unintended receivers and attackers, whatever strength is inherent in the algorithm is compromised and the confidentiality of data encrypted with the key cannot be guaranteed. Thus, when considering a symmetric key encryption scheme, it is equally important to evaluate the key transfer mechanism. For example, a mythical algorithm that is unbreakably strong and uses a key that has so many combinations that brute-force attacks are infeasible is compromised if the keys are transferred over the phone or via postal or electronic mail.

Figure 1 illustrates the encryption process using a symmetric key cipher. Sometimes other values are provided to the encryption algorithm for initialization purposes. The resulting ciphertext will bear no relation to the plaintext. Figure 2 shows how decryption is accomplished by reversing the process. If values other than the key were used to initialize the encryption operation, they are required inputs to the decryption algorithm. The resulting plaintext will be a faithful reproduction of the original plaintext. Using the wrong key in the decryption process, even if different from the correct key by just one bit, results in meaningless output.

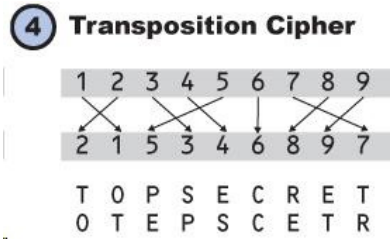



Fortunately, the millennia-old necessity of securely transferring a secret key from a sender to a receiver has resulted in some ingenious methods. Along the way, it was also determined that if the key was only used once then destroyed, the resulting system, called a *one-time pad,* is mathematically proven to be unbreakable through cryptanalysis. Naturally, a lot more keys have to be transferred when using a one-time pad and the keys still need to be distributed in a secure manner, so this is not commercially feasible.

Symmetric encryption algorithms are primarily used for bulk encryption of data, such as an entire file, document, or bundle of transaction data. The two fundamental symmetric encryption techniques are *substitution* and *transposition.* Substitution ciphers are simple and operate by *replacing* each character with another character, for example, the letter 'a' would be substituted for the letter 'g' every place it occurs. Substitution ciphers are rarely used today due to the ease in breaking them with frequency cryptanalysis, in which the frequency of encrypted characters in the ciphertext is used to derive the plaintext. Figure 3 is an example of a substitution cipher.




In contrast, transposition ciphers operate by moving plaintext characters to new locations in the ciphertext, rather than by substituting individual characters. An example of a simple transposition cipher is the word jumble or cryptogram in a newspaper. All the characters found in the plaintext are in the ciphertext, but in different relative positions. Unlike a word jumble, which is a random transposition, transposition-based encryption works by moving characters around in a definite pattern that is reversed to decrypt the ciphertext. Pure transposition ciphers are not used in modern cryptography because of the ease of computer-based cryptanalysis. Figure 4 is an example of a transposition cipher. The key for such a cipher is a representation for the character replacement scheme.




The principles of substitution and transposition are, however, combined into diffusion ciphers, which are used for all modern symmetric key ciphers. Diffusion algorithms not only substitute differing values for the plaintext characters, but also spread the characters throughout the ciphertext. A significant strength of many diffusion-based algorithms is that the same character will actually be encrypted into a different symbol based on its location in the plaintext and the data that precedes it.

The best secret key algorithms possess a property known as the Avalanche effect, in which even a one-bit change in the plaintext results in changes in approximately one-half of all the ciphertext bits. Symmetric ciphers are fast and typically compact in terms of their computer code size and memory requirements, which is important as encryption capabilities are extended to devices like PDAs and smart phones that have power, processor, and memory limitations.

Symmetric algorithms can be further divided into *block* and *stream* ciphers. Block algorithms encrypt and decrypt a fixed-size block of cleartext and ciphertext, respectively, usually a multiple of 64 bits. Stream ciphers, on the other hand, continuously encrypt any amount of data as it is presented, usually by mathematically combining the data with a *keystream,* an infinitely long key sequence that is generated based on a finite key starting value.[6] Following are examples of both types of algorithms.

**Common Symmetric Ciphers**

The following are the secret key algorithms most commonly used today, listed in chronological order of their inception.

**Data Encryption Standard (DES)**
DES is a standardized and published encryption algorithm, approved by the U.S. Government in 1977 after considerable analysis. The genesis of DES is traced back to a cipher termed Lucifer, invented by Horst Feistel of IBM. It uses a 56-bit key, which is sometimes stored with additional parity[7] bits, extending its length to 64 bits. DES is a block cipher and encrypts and decrypts 64-bit data blocks. Although at the time of its inception, the effort to crack a 56-bit key was considered so enormous as to prevent brute-force attacks, it is now considered insecure, and all government agencies must use algorithms with longer keys, as discussed below. Despite the obsolescence of DES due to its key length, it is quite elegant and the most cryptanalyzed algorithm in the world, withstanding all attacks on the algorithm itself.

**RC4**
RC4 is a stream cipher, also created in 1987, and its only complexity is in the generation of the keystream, which is potentially an infinitely long sequence of key values, which start with a 40- or 128-bit key, and a 24-bit initialization vector (IV). The actual encryption step is very simple; the keystream is combined with the plaintext in an XOR[8] operation. Using the same key and IV, the keystream is totally reproducible, so in practice the sender and receiver using this algorithm will each be generating an identical keystream. RC4 is ten times faster than DES.

**RC5**
RC5 is a fast, parameterized block cipher, with a variable block size (32, 64 and 128 bits), variable key size (0 to 2040 bits), and a variable number of rounds (0 to 255), or individual encryption steps.[9] RC5 is patented by RSA.[10] It can be used as a drop-in replacement for DES, with the block size set to 64 bits and the key size set to 56 bits.

**Triple DES (3DES)**
Triple DES is simply three successive encryptions with DES.[11] It is possible to use either two or three distinct keys with 3DES. Thus, for the three-key case, one obtains the benefit of a 168-bit key space[12] with the known strength of the DES algorithm. Performed correctly, 3DES is as unbreakable a secret-key algorithm as any known, but it is slow. 3DES is defined as ANSI standard X9.52, and has been widely used in commerce and

government applications.

**Advanced Encryption Standard (AES)**

The National Institute of Standards and Technology (NIST) selected an algorithm called "Rjindael"[13] on October 2, 2000 as the AES in a multi-year competition. AES replaced DES. AES is projected to provide secure encryption of sensitive but unclassified government information until 2020. Rjindael is a fast block cipher, with variable key length and block sizes (each can be independently set to 128, 192 or 256 bits). AES became an official U.S. Government standard in 2002. Like DES before it, AES is now widely used for commercial and private encryption purposes. One significant benefit of AES is that the algorithm is public, and its use is unrestricted, with no royalties or license fees owed to the inventors or the government.

**Public Key Encryption**

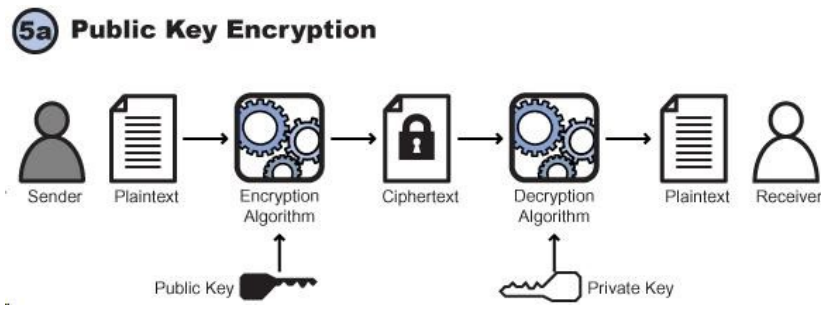
As discussed above, a significant drawback to secret key encryption is the requirement of securely distributing the shared key to intended receivers. The fact that for three thousand years the problem of securely transferring keys was unsolved was certainly not due to lack of effort. It had long been recognized that what was needed was a key exchange system in which the parties could exchange some key-like numbers, which would then permit each person to independently derive the secret key, where an unintended receiver could intercept the exchanged values without being able to also derive the secret key.

What was missing was the mathematical insight to solve this pressing problem. In 1976, Hellman hit upon the application of modular arithmetic in a simple scheme that, for the first time in history, allowed two parties to independently derive the same secret key without needing to transfer confidential information. Finally, the key distribution problem had been solved, a watershed event in cryptography. This key distribution method, known as Diffie-Hellman (DH) Key Exchange, is still used today.

Although Diffie-Hellman was a spectacular breakthrough, the iterative nature of the key derivation process was inconvenient. Generating a new secret key required the active participation of both sender and receiver. The search continued for a one-way function that would breathe life into the theoretical principle of public key encryption.

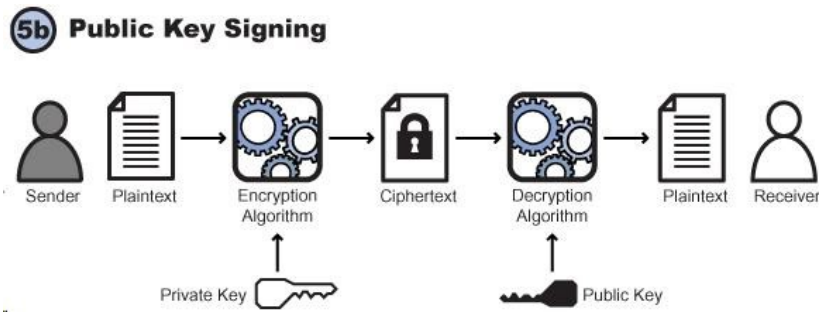
It was made by Ronald Rivest, Ari Shamir and Leonard Adleman, all researchers at MIT and inventors of the public key encryption algorithm called RSA.[14] RSA not only eliminated the need to transfer secret keys, but also facilitated convenient and efficient encryption by removing the Diffie-Hellman requirement of exchanging values back and forth.

Figure 5a demonstrates how RSA works. Note that the public and private keys referenced in the figure are part of the receiver's *key pair.* When the sender wishes to encrypt information that only the receiver can decrypt, she uses the receiver's public key to encrypt. The public key, as the name suggests, can be freely distributed in the clear. It can be sent via electronic or postal mail, posted on a billboard, or spoken over the telephone without sacrificing any security. It is essential, however, that the private key be kept inviolable and never shared or divulged to anyone.




Note the fundamental differences between asymmetric and symmetric key encryption. When using secret key ciphers, there is a different secret key for each pair of parties communicating. In the public key case, there is just one key pair for each receiver, because the public key can be distributed to everyone who wants to send encrypted data to the receiver. Having the public key allows senders to encrypt data, but without the private key, they are unable to use the public key to decrypt communications from anyone else using the same key pair.

Equally important as the advantages inherent in public key encryption is the support for the properties of *authentication* (identification of the sender) and *sender non-repudiation* (the inability of a sender to refute that they signed something encrypted with their private key). Since anyone with the sender's public key can decrypt a message encrypted by the sender's private key, this type of encryption, called a *digital signature,* does not protect the confidentiality of the message. The sender is prevented, however, from denying that he was the originator of the information thus signed, unless the private key was compromised. Figure 5b illustrates the operation of signing.




Here, the sender's key pair used to create the digital signature. Although technically the operation performed is still encryption and decryption, the operations are termed *signing* and *verifying,* respectively. Because assuring the privacy of data is almost always important, it is rare that something will be signed but not encrypted. When information is both encrypted and digitally signed, it is wasteful to sign the entire document or message, as that would result in the sender transmitting twice the amount of information actually needed. Therefore, it is most common to create a signature of a value termed a *message digest* or *hash,* which is a compact value created by a message digest or hash function that represents the plaintext.

Message digests are fixed in size and small, usually 128 or 160 bits. The message digest will change dramatically if even one bit of the plaintext changes. The combination of digitally signing a message digest supports *integrity checking* and sender non-repudiation. Here's how it works. Before the data is encrypted, a message digest is computed of the plaintext. The data is then encrypted with the receiver's public key. The message digest is signed with the sender's private key and the resulting signed hash appended to the encrypted data. Another permutation is to then compute a hash over the ciphertext and possibly sign that as well. When the data is received, the receiver verifies the signature by decrypting the hash with the sender's public key. The receiver's private key is then used to decrypt the data, and immediately the message digest is computed over the plaintext. If the verified digest does not match the computed digest, it means one of two things. The information changed in transit, indicated that integrity has been compromised, or that the sender is not who he or she purports to be. In either of those cases, the data is not reliable and is discarded.

The difficulty in locating one-way functions suitable for public key cryptography has resulted in few algorithms in use. Let's examine the most prevalent algorithm, RSA.

**The RSA Public Key Algorithm**

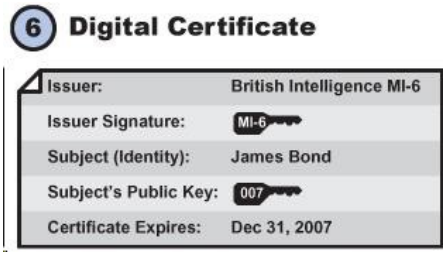
RSA is the undisputed leader in public key encryption. The algorithm's one-way function is based on the *intractability,* or mathematical difficulty, of factoring the product of two prime numbers.[15] In RSA, the product of the prime numbers is the public key, and the two prime numbers make up the private key. If an attacker can factor the public key, the private key is thus compromised, and it is possible to decrypt information encrypted with the public key.

With all the benefits of RSA public key encryption, one might wonder why it isn't used to encrypt everything, rendering symmetric key algorithms obsolete. The reason is speed. The fastest public key operation is significantly slower than the slowest symmetric cipher, such as Triple DES, making RSA unsuitable for bulk encryption. RSA is quite useful for encrypting a secret key in a *key envelope,* for secure transmission to the intended receiver of information. RSA supports authentication, sender non-repudiation and data integrity, which cannot be provided by symmetric algorithms alone. The fundamental strength of RSA is well-established; the only attacks on it are by factoring the public key or mounting a brute-force attack, the latter being particularly hopeless given the extraordinarily large key space of a 1024- or 2048-bit key. As computers are able to factor numbers faster, the minimum recommended RSA key size will continue to increase.

**Public Key Management**

The many advantages of public key encryption do not come without complexities as well. The need to store and distribute public keys in an efficient, scalable way is a significant challenge. Ideally, public keys should be easily searchable and retrievable as the need to send encrypted information arises. This is the function of Public Key Infrastructure (PKI), which is beyond the scope of this article. However, we will discuss a crucial component of PKI, the *digital certificate.*

Figure 6 shows a simplified digital certificate. It is essentially an electronic identification card that can be used to identify a person, company, or a computer. X.509 digital certificates are the most common. Digital certificates are issued by trusted entities known as Certificate Authorities, charged with vouching for the identity of the certificate holder. Embedded in the certificate is the public key of the *subject* of the certificate, which can be extracted and used to encrypt information that only the subject of the certificate can decrypt.



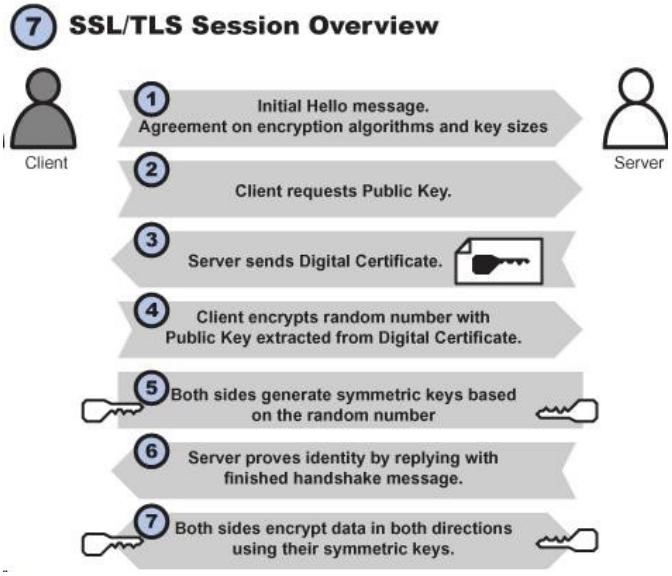

How does one know that the certificate is really issued by the purported issuer? The certificate contains the digital signature of the issuer. Recalling that the issuer created this signature with its private key, we need to obtain the public key of the issuer in order to verify the signature and ensure that the certificate truly originated from the issuer. The issuer's public key is located within yet another certificate called the *root certificate* for that

issuer, which is self-signed by the issuer. Once the authenticity of the certificate is determined by verifying the signature, the subject's public key is extracted, providing the certificate is not expired or revoked, and the information destined for the subject is encrypted, and perhaps signed, and sent to the subject. The importance of digital certificates will be highlighted below.

**Applications of Encryption**

Having explored the operation and traits of both symmetric and asymmetric encryption algorithms, let's see how they are both used in the Secure Sockets Layer (SSL) protocol.

**Secure Sockets Layer (SSL)**
SSL is the de facto Internet security standard. Although almost always used to secure World Wide Web (WWW) sessions and transactions, it is technically applicable to other forms of TCP/IP [16] network communications, such as the File Transfer Protocol (FTP). Whenever a Web site is accessed with the https:// protocol, SSL is automatically invoked. SSL uses a combination of symmetric and asymmetric key algorithms and message digest functions to protect the confidentiality and integrity of data sent to and received from a Web server, and to authenticate the server's identity. Figure 7 illustrates the initial connection and handshake process that occurs each time a secure Web site is accessed. [17]




In the first step, the client computer connects to the secure Web site and requests an encrypted session. The two computers negotiate the strongest cryptographic suite supported in common on both systems, which is a combination of various symmetric algorithms and key sizes. Once a compatible cryptographic suite is negotiated, the client requests the server's public key. The server responds with its X.509 digital certificate, which contains the server's public key. The client generates a pseudo-random number called the *premaster secret* with its PRNG, encrypts it with the server's public key, and sends the encrypted value to the server. The server decrypts the premaster secret and the client and server each process the premaster secret to create a *master secret*. The master secret is then used to create three distinct keys on the client and server, one of which serves as the symmetric key for the SSL session. The client sends a message indicating that all further messages will be encrypted, and the server responds with a similar message. All further communications are encrypted in both directions until the session ends.

SSL is perhaps the most successful security protocol ever implemented. The Internet Engineering Task Force (IETF), which is charged with managing the technical direction of the Internet, drafted the open-standard replacement for SSL, called Transport Layer Security (TLS), now version 1.2. It is modeled directly after SSL, which was created by Netscape, and has implemented enhancements to SSL.

**Conclusion**

The importance of computers and networks and the information they store and communicate to society today are equaled only by the threats to them. The recent departure of Google from mainland China over a widely-publicized attack there on its e-mail system is an ominous reminder of the growing attacks on data and communication networks. The encryption algorithms discussed in this article are in many instances the only protection between our critical information and those who seek to compromise and exploit it.

**Footnotes**

1. See David Kahn, "The Code Breakers: The Story of Secret Writing," Simon & Schuster, 1996.
2. For a fascinating and entertaining survey of the role cryptography has played in history, see Simon Singh, "The Code Book," Anchor Books, 1999.
3. Some algorithms have a third input called an Initialization Vector (IV) that can be transmitted in the clear, unlike the key, which must always be protected. Knowing only the ciphertext and IV requires an exhaustive key search.
4. In practice a brute force attack requires on average that one-half the possible keys be tried.
5. Examples of these values are date, time, and process identifier.
6. Stream ciphers commonly require both a starting key and an initialization vector (IV), as mentioned supra.
7. Parity is a primitive but effective method of detecting key integrity problems, which might indicate tampering or that errors were introduced in transfer or storage.

8. XOR stands for 憁xclusive OR and is a standard logical operation performed on two values on a bit-wise basis. If one value is 0 and the other 1, the XOR output is '1,' whereas the XOR output is 0 if either both inputs are 0 or 1. The XOR operation is not only extremely fast, but has the useful property of being symmetric. For example, if the first four bits of the keystream and plaintext are, respectively, 1011 and 0010, the result of XORing them is 1001, the ciphertext. Notice how decryption works: the keystream is identical, so 1011 and 1001 are XORed, resulting in 0010, the original plaintext.
9. Increasing the number of rounds increases the time it takes to encrypt and decrypt, but it also makes breaking the ciphertext through linear and differential cryptanalysis more difficult. It is estimated that RC5 is immune from cryptanalysis when 16 or more rounds are used.
10. U.S. Patent No. 5,724,428, issued March 3, 1998 to Ronald Rivest, assigned to RSA Data Security, Inc.
11. In practice, Triple DES is typically used in EDE mode, meaning the data is first encrypted using key K1, then decrypted using key K2, then encrypted once more with key K3.
12. One might think that a brute-force attack would take 3 times $2^{56}$ iterations, but because theoretically at least it is necessary to try all key combinations, in the worst case a brute-force attack would require $2^{168}$ keys to be tried, an extraordinarily large number (the estimated number of atoms in our planet is $2^{170}$).
13. Rjindael (pronounced "Rhine Doll") is the creation of two Belgian cryptographers, Dr. Joan Daemen and Dr. Vincent Rijmen.
14. 'RSA' is an acronym comprised of the first letter of the inventors last names. See U.S. Patent No. 4,405,829, issued September 20, 1983 to Ronald Rivest, Adi Shamir and Leonard Adleman, and assigned to the Massachusetts Institute of Technology (MIT). MIT granted an exclusive license to RSA Security Inc. On September 6, 2000 RSA disclaimed the remaining two weeks of the patent term, placing it into the public domain.
15. Prime numbers are those numbers only divisible by themselves and 1, such as 1, 3, 5, 7, 11, 13 and so on.
16. TCP/IP is the Transmission Control Protocol/Internet Protocol, the network protocol that defines how computers communicate over the Internet.
17. In this context, a secure Web site is one that supports SSL or TLS and has the appropriate digital certificate installed. These two traits do not necessarily imply that the Web server as a whole is secure.

**Additional Reading on Cryptography**

To learn more about cryptography, visit Understanding and Applying Cryptography and Data Security

**About the Author**

Barry K. Shelton is a registered patent attorney and patent litigator with the law firm of Bracewell & Giuliani, LLP, and formerly an electrical engineer. His technical interests include cryptography, network security, and data compression. He holds a M.S. in Electrical Engineering from San Diego State University and a Juris Doctor from the University of San Diego School of Law. He is a registered professional engineer in California and a member of the IEEE. He can be reached at barry.shelton@bgllp.com.