Encryption in The Digital Age

Paul S. Wang, Sofpower.com

September 5, 2021

The Internet and the Web made the entire world a global village. Digital information travels at light speed making instant communication and interactions among people, near and far, a reality. Yet, because of their open nature, data traveling on the Internet and Web are subject to public view, making information security and privacy a real concern for almost everyone.

What is done to safeguard our sensitive information in the digital village? What can individuals do to protect personal information? In a previous post of my *Computational Thinking* blog (`computize.org/ctblog`) "*Cyber Security—How Not to Be A Fish*" we talked about many useful topics. Here we will focus on *cryptography*, one of the most important techniques to understand.

Egg Scrambling

Cryptosystems keep communication safe by encryption, a technique invented long before digital computers. The concept is simple—an original message (*plaintext*) is encrypted into *ciphertext* before communication. The ciphertext is gibberish for anyone except the intended receiver who knows how to decrypt the ciphertext back into the original plaintext.

All this sounds abstract but it is really straightforward—you scramble a message, like scrambling an egg, so it becomes garbled. There is no way for anyone to find out what the original egg looks like by looking at the scrambled egg.

Hashing is a systematic way of scrambling any message of arbitrary length into a fixed-length sequence of bits. The same original message becomes the same garbled *hash* or *message digest* after hashing. The slightest change in the original message results in a very different hash. Given a hash, there is

no way to know what the original message was. There are quite a few hash algorithms. SHA-256 is currently used most often.

For example, the message "This is the egg." has the following SHA-256 hash (a sequence of 256 bits), displayed in HEX (base 16) notation:

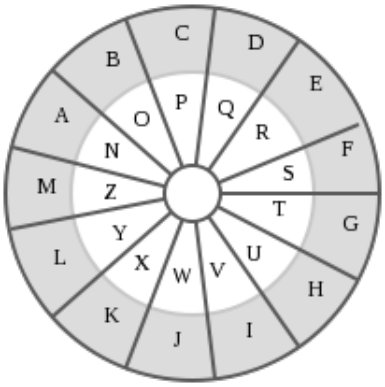
a6d39dfb1c20cd15b1a69506dd072354cb4c472135ccc51a0412f42d64e6e415

In fact this is exactly how your password is stored on the server where you login. When you set your password it is first hashed and then the resulting hash stored instead. By not storing your actual password its secrecy is preserved. When you login, the password you enter is hashed the same way and the result compared to the stored version. A match means your password is correct. Thus, hashing is a *one way* encryption, there is no way to recover the original message from the hash.

Still, a method is needed to protect your password from eavesdropping when you login online, that leads us to secure communication.

Secure Communication

For secure communication, we need a system where the encryption can be undone or decrypted. For example, *rot13* encrypts by simple letter substitution. Each letter in the plaintext is replaced by a letter 13 places after it, assuming there are only 26 letters and the last letter is followed by the first letter in a cycle. A rot13 ciphertext can be decrypted by applying the encryption on it again.



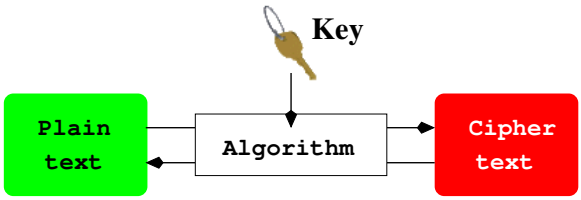
For example, the plaintext "Now I know my ABC" becomes the ciphertext "Abj V xabj zl NOP" using rot13. It can be entertaining to decrypt it yourself.

It is not hard to think of other more sophisticated ways. For example, senders and receivers can agree on a book to use. Ciphertext would contain page number, line number and word number to identify any particular word from the book. Only people who know which book and what the numbers mean can decrypt the message. Further, one of the many numbers may indicate a particular book among several possible ones to use.

These are examples of *symmetric cryptosystems* that use the same *key* to encipher and decipher messages. Communicating parties must know the key beforehand. And the key must be kept secret to others. Obviously, rot13, the book plus numbering scheme, and the WW2 German Enigma machine settings are the keys.

Symmetric Cryptosystems

Modern symmetric encryption systems work on digital data. Most, if not all, of them use an encryption/decryption algorithm that is open and a key that is kept secret.



- Encryption/decryption algorithm: The algorithm performs various substitutions and permutations on chunks, typically 128- or 256-bit blocks, of the plaintext or ciphertext.
- Secret key: The plaintext (ciphertext) and the secret key are input to the encryption (decryption) algorithm. The exact transformations performed depend on the key used. The algorithm produces differ-

ent output depending on the key given. Using the same key on the ciphertext, the algorithm produces the original plaintext.

A symmetric cryptosystem usually has these two characteristics:

1. Open algorithm: The encryption/decryption algorithm can be described in the open. This works because it is impractical to decode any ciphertext knowing the algorithm and not the secret key.

2. Secret key: Senders and receivers must have obtained the key securely in advance and must all keep the key secret.

The secret key is usually a bit pattern of sufficient length. The quality of the secret key is important. It should be randomly generated and 256-bit or longer to make brute-force attacks, trying all possible keys, impractical.

When a password (or passphrase) is used as a key, it is usually put through a key derivation function, which compresses or expands it to the key length desired. Often, a randomly generated piece of data, called a *salt*, is also added to the password or passphrase before transforming it to the actual key.

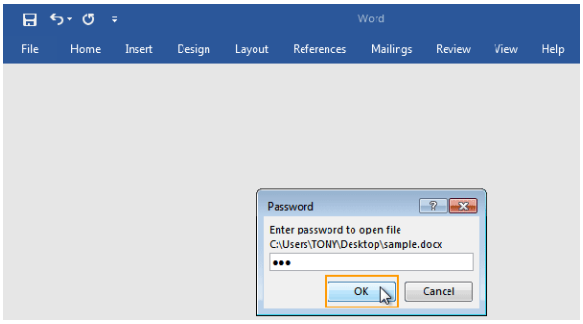
The Advanced Encryption Standard (AES) is a symmetric cryptosystem for digital data established by the US National Institute of Standards and Technology (NIST) in 2001. AES has been adopted by the US government and is now used worldwide. It supersedes the Data Encryption Standard (DES), which was published in 1977. There are other symmetric ciphers, such as RC4 and Blowfish, but AES-256 seems to be the best.

Keeping Files Safe

Let's apply computational thinking and use encryption to keep our own sensitive files safe.

Files of bank statements, login information, business contacts, tax returns, accounting, financial, and insurance records, contracts, medical history, and so on should be kept confidential by encrypting them. Furthermore, it is advisable to keep scanned images of your important documents, such as birth certificates, passports, and driver licenses, in files for easy usage. But, make sure these are protected, too, by encryption.

With Microsoft Word™or Microsoft Office™, you can save and retrieve encrypted files.



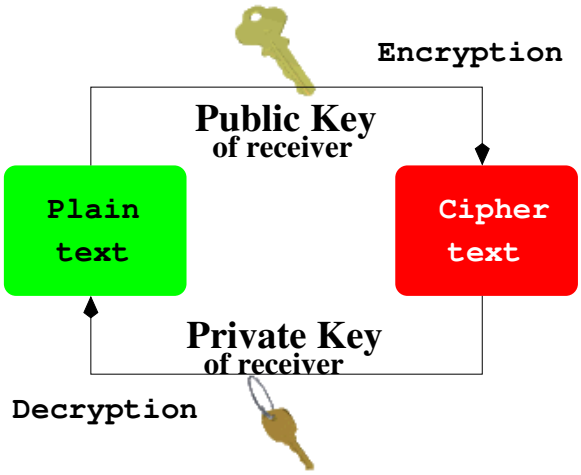
The free Vim editor (`vim.org`) can encrypt/decrypt a file using Blowfish and a secret key. The application 'AES Crypt' (`aescrypt.com`) simply encrypts/decrypts a file with AES using a key that you choose. These tools work on multiple platforms. Similar file encryption apps are available on smartphones.

Recording all your passwords and secret keys in an encrypted *keyfile* means you can access login information and secret keys if you just remember one key, the one for the keyfile. This can make life much easier. You may even consider encrypting your hard disks.

Symmetric cryptography is also used in HTTPS, the secure Web protocol, to keep communication between a user and a website confidential. HTTPS is a must for any site requiring login. But good practice now calls for all websites to use HTTPS, instead of HTTP, in general. Every time you visit a website under HTTPS, a *session key* is agreed upon **automatically and securely** between your browser and the website first. For this and some other purposes, *public-key cryptography* is used. Please read on.

Public-key Cryptosystems

Unlike symmetric systems where the same key is used for both encryption and decryption, *Public-key cryptosystems* are asymmetric and use not one but a pair of keys—one to encrypt and the other to decrypt. The decryption key is kept secret, while the encryption key can be shared publicly.



The pair of keys are integers satisfying well-defined mathematical properties and usually produced by a key generation program. For each public key, there is only one corresponding private key, and vice versa.

The public key is made available for anyone who wishes to send an encrypted message to a recipient who uses the private key to decrypt the message. It goes without saying that the owner of the key pair will keep the private key safe and secret.

The public key usually becomes part of a *digital certificate*, which can be presented to anyone to verifiably associate the key to its owner. An owner of a key pair may also place the public key in online *key repositories* open to the public.

Thus, anyone who wishes to send a secure message to a particular receiver will use the receiver's public key for encryption. The receiver can then use the corresponding private key for decryption. Note that in this case the parties need not have communicated with each other ever before.

For example, if Eve wished to send a secure message to Adam whom she had never met before, she would first look up his public key, encrypt the message with it, then send the result to Adam.

Thinking Outside of The Box

One aspect of CT is *to break out of the mold of conventional thinking, to challenge old assumptions, to think outside of the box.*

Public-key cryptography is a great example. It started with the breakthrough idea, credited to Bailey W. Diffie, Martin E. Hellman, and Ralph C. Merkle, that *two parties can establish a secret symmetric key over an unprotected public communications channel*. It is infeasible for any eavesdropper of their open exchanges to deduce the key they agreed upon.

When Merkle first proposed research into the idea, it was met with skepticism and resistance from experts. But, this mid-1970 invention soon led to the even more astounding public-key cryptography, where the encryption algorithm and key can be published openly.

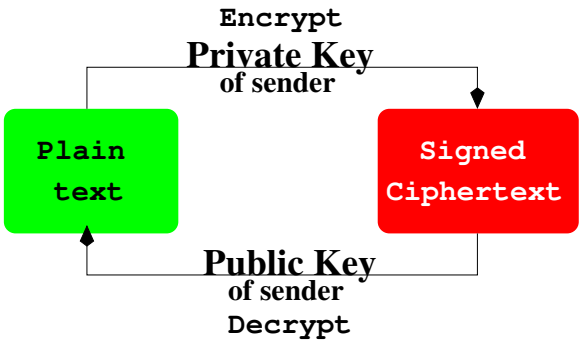
One critical element of public-key cryptography is the computational infeasibility to deduce the private key from the public key. Unlike symmetric encryption, public-key encryption avoids communicating secret keys. Thus, anyone and any system can send a secure message to a receiver, who has a published public key or a digital certificate, without having any prior contact with the person.

However, the speed of public cryptography is much slower than symmetric cryptography. This explains why HTTPS uses public-key encryption only in the handshake phase to establish a symmetric *session key* for the actual data transmissions.

Public-key cryptography underpins modern cryptosystems, applications and protocols, such as digital certificates and HTTPS. It also can achieve *digital signature*.

Digital Signature

Digital signature is a way to use encryption to transform a document digitally into a signed document such that (1) the signed document cannot be altered in any way, and (2) everyone can verify it has been signed by the signer. Digital signature is different from electronic signature. The latter is simply an electronic form of a person's autograph.



A public-key cryptosystem lends itself immediately for digital signature. To sign a document, a signer simply encrypt it with the signer's own private key. The resulting document can be sent to any interested party. The signed document can be decrypted using the signer's public key. Because only the signer has knowledge and use of his/her private key, the document is thus verifiably signed by him/her. Furthermore, the integrity (unaltered property) of the original document is also maintained. Failing to protect your private key can be serious. Someone may use it to sign stuff and get you in trouble.

For example, if Eve had a key pair and wished to send a signed contract to Adam, she would use her own secret key to encrypt the contract, then send the result to Adam. The fact that Adam, or anyone else for that matter, can decrypt the result with Eve's public key to reproduce the contract proves that she indeed had signed it. Furthermore,it also shows the contract is unaltered because any alteration will result in decryption failure.

Normally the contract is no secret. Therefore, Eve may digitally sign an SHA-256 hash of the contract instead.

Applications

Cryptosytems can be complicated and hard to understand all the details. But they basically use clever algorithms to scramble and unscramble messages to achieve secrecy and security in many important applications. In fact, we can say that without cryptography protection of sensitive data in the digital world would be very hard indeed.

We can list some applications.

- Safe password storage and checking via hashing.

- Encryption/decryption in Microsoft Word™, PDF and other textual files.
- Direct file encryption/decryption with AES.
- Software distribution with digitally signed message digest.
- Digital certificates with owner's public key.
- Symmetric and asymmetric cryptography used in protocols and apps such as SSH (secure remote login), SFTP (secure file transfer), SSL/TLS (transfer layer security), and HTTPS (secure web protocol).
- Secure email with PGP (pretty good privacy), GPG (Gnome privacy guard), and products such asMS Office/Office365™.
- Digitally signed contracts and documents, digital certificates being widely used examples.
- Hashing and digital signature for transaction integrity in blockchains.
- Cryptography in digital currency.

Finally

The digital age brings so much power and convenience to all of us. It makes the entire world a global village. That also means all kinds of hackers and dangers lurk online. Cryptography is the weapon of choice to defend our security and privacy in the digital world. We owe it to ourselves to understand and apply it well.

Hashing, symmetric encryption/decryption, public-key cryptography, digital signature, digital certificates, and secure protocols such as HTTPS combine to form a digital security infrastructure designed to keep our files and data safe and secure. Various apps, such as SSH or MS Office/Office365™, support security features based on cryptography. As users, we can select the right applications to safeguard our private information.