Cryptography - The myths

You may wonder why cryptography is so important and the people need to study more about it. We will learn more about the cryptography and the awareness that every information specialist should know about cryptography and its importance.

Cryptography - is the study of mathematical techniques related to the aspects of information security such as confidentiality, data integrity, authentication and data origination.

We are going to see the history and science of cryptography. The science behind every form of security, authentication mechanisms, information or data safety, in our words, *information security* and what not. The reason is, whichever operating system you use, whatever programs or authentication systems you deploy, the actual strength of the system to withstand attacks against possible ways depends highly on the underlying cryptosystem.

The importance for information security has an old history. The story begins when *Julius Caeser* send messages to his trusted acquaintances, he didn't trust the messengers or likely to know that message can be intercepted while on the way. So he replaced every 'A' with 'D' and 'B' with 'E' and so on through the alphabets. Only someone who knew '**SHIFT BY 3**' could ***decipher*** his message.

For example,
if the original message was,
'GIVETWOMILLION' ➔ **PlainText**
he would have encoded the message with 'Shift by 3' and produced the message as
'JLYHWZRPLOORQ'➔ **CipherText**
which obviously is in an unreadable format unless you know the method of ***deciphering***.

Some of the common terms that are used in cryptosystems are explained here. The original message is called as the ***plaintext***. The disguised message is called as the ***ciphertext***. The method of producing *ciphertext* from *plaintext* using the key is called as ***encryption*** or ***enciphering*** or ***encoding*** and the reverse procedure of producing the *plaintext* from *ciphertext* using the key is called as ***decryption*** or ***deciphering*** or ***decoding***. The people who are supposed to receive the disguised message are called the ***recipients***, and other people are ***enemies, oppenents, interloppers, eavesdroppers, tappers*** or ***third parties***. A ***cryptosystem*** means a collection of algorithms. Algorithms are labelled and the labels are called the keys. For example, if caeser used, '*SHIFT BY n*', algorithm means, *n* is the ***key***. Key plays and important role the cryptosystem because the whole strength of the algorithm used depends on how the key is choosed. For a very strong algorithm, and an easily guessable key means, the encoding of the message goes as a waste.

The science of breaking ***cryptosystems*** is called the ***Cryptanalysis***. Though cryptanalysis may not be the easier to learn, because of it complexity invovled in breaking the ciphered messages, but to decipher it ***without*** the knowledge of the cryptoalgorithm and the key used. The study of both cryptography and cryptanalysis collectively is called as the cryptology. It may be possible to find the key used or to find the plaintext from the ciphertext, by a cryptoanalyst by using various attack techniques.

Cryptanalysis plays an important role in the cryptography because, it attacks the encoded message to produce the relevant plaintext. For example, if a function f(x) gives, y and you know the function and the result y, all you need to find is the key x, which can be found using many methods. One such method is called as ***brute-forcing.*** It is nothing but substituting all possible values of x in the function f and match the output with y, the one that matches, gives the x which is also the key.

The all possible values of x means the ***keyspace*** where the key is searched. For every known algorithm today, there exists a *keyspace* and the technique of searching called as the ***keysearch.*** You may think then, every know algorithm can be broken, but the complexity of the algorithm and the keystrength, makes the task tougher. The massive amount of computing power and time required to break the message and to find the key makes the algorithm stronger. A good cryptosystem, should be able to withstand different kind of attacks, depending upon the algorithm used. Various other methods used to break the encrypted messages are classified attacks, interesting combination of analytical reasoning, application of mathematical tools, pattern finding, patience, determination, computing power and offcourse not the least but luck. Also the computational number theory posses the threat to what so called as the public-key crypto systems, which are widely used in all formats in modern secure transactions, authentication and digital certificates. Though the keyspace and amount of computational power makes it impratical for such attacks.

Why Cryptography?

The main use of cryptography is to provide the following as mentioned earlier.

(1) privacy or confidentiality
(2) data integrity
(3) authentication and
(4) non-repudiation.

1. ***Confidentiality*** is a service used to keep the content of information from all but those authorized to posses it. *Secrecy* is a term synonymous with confidentiality and privacy. There are numerous approaches to providing confidentiality, ranging from physical protection to mathematical algorithms which render data unintelligible.

2. ***Data integrity*** is a service which addresses the unauthorized alteration of data. To assure data integrity, one must have the ability to detect data manipulation by unau-thorized parties. Data manipulation includes such things as insertion, deletion, and substitution.

3. ***Authentication*** is a service related to identification. This function applies to both enti-ties and information itself. Two parties entering into a communication should identify each other. Information delivered over a channel should be authenticated as to origin, date of origin, data content, time sent, etc. For these reasons this aspect of cryptog-raphy is usually subdivided into two major classes: *entityauthentication* and *dataorigin authentatio*n. Data origin authentication implicitly provides data integrity (for if a message is modified, the source has changed).

4. ***Non-repudiation*** is a service which prevents an entity from denying previous commitments or actions. When disputes arise due to an entity denying that certain actions were taken, a means to resolve the situation is necessary. For example, one entity may authorize the purchase of property by another entity and later deny such autho-rization was granted. A procedure involving a trusted third party is needed to resolve the dispute.

A fundamental goal of cryptography is to adequately address these four areas in both theory and practice. Cryptography is about the prevention and detection of cheating and other malicious activities and to secure what you have as sensitive information.

Various avatars of cryptography.

The various methods that are devised with relevant to time starting from historic events. It is very tough to spot the origin of cryptography, because it started when men, want to hide or protect the information, to provide confidentiality, it became a practice.

Julius caeser, used the old method of shifting for producing ciphertext, which is called as the ***simple monoalphabetic substitution cipher***.

Then various modified versions of the same shifting became a practice in *usenets*, where the message if shifted by 13 (**rot13**), which on encoding twice produces the plaintext. Try it out yourself as there are 26 alphabets.. It was the one of trivial encodings used for a long time on message boards and news servers. So the news readers have builtin **rot13** functions to decode or encode plaintexts.

For your assistance I include the relevant substitution table.

A B C D E F G H I J K L M
N O P Q R S T U V W X Y Z
N O P Q R S T U V W X Y Z
A B C D E F G H I J K L M

The next was the ***atbash*** system, which is a modified version of the caeser and even easier to solve. The only difference was *Caesers* was roman in origin and *atbash* was jewish in origin. In the atbash system, the last letter represents the first letter and first letter represents the last letter and so on. So there was pratically only one solutions and quickly decipherable. The table that was used in atbash is

A B C D E F G H I J K L M
W V U T S R Q P O N M L K
N O P Q R S T U V W X Y Z
J I H G F E D C B A Z Y X

The combination of caesers and atbash was used in some encodings schemes.

The other methods are the monoaplhabetic substitution where a letter of plaintext always produces the same cipher text. The usage is similar to atbash and caesers, but the order of the letters is user defined. So you can produce a ciphering table for a monoalpabetic cipher. It was cracked by the methods known as frequency search, which searchs for vowels which occur more in an paragraph and concluding the keys for that alphabet. The famous phrase used was ***'pen and paper techniques'*****,** which required a little common sense and analytical reasoning for breaking the substitution ciphers.

Various other methods like ***Polybius Chequerboard, Vigenegravere or the famous autokey cipher, Playfair, Transpositional cipher****,* etc*,* came into picture. They collectively used key to manipulate the plaintext, which was broken easily in many cases. But there are some historic ciphers which remains still un-deciphered.

> With the invention of crypto machine called ***enigma*** by the allies and germans, during the World War I and World War II for the transfer of commands and orders to the troops and about the actual situation. Some people intercepted the messages but failed to decrypt due to the constant improvement and change done with the machine architecture.

Modern Cryptography

The classification divides itself as the method of enciphering the plaintext to ciphertext differs into various methods that are devised. The cyptography branches itself into ***Symmetric Key cryptography often known as conventional cryptogrphy or secret-key cryptography*** *and* ***public-key cryptography (asymmetric key cryptography) or public/secret pair cryptography***. The *symmetric key* technique further classifies itself into ***Block ciphers*** *and* ***Stream ciphers.***

Symmetric key cryptography is the method where the plaintext is converted to cipher text based on the ***unique key*** and the function used. The actual strength of the symmetric key cryptography lies in choosing the non-reversible function that uses the key to produce the cipher text. Like a function

$$\mathbf{E_K(P) = C,}$$

where E is the function and **K** is the key used and **P** is the plaintext and **C** is the ciphertext produced. Then again the use of function

$$\mathbf{D_K(C) = P,}$$

where D is the function used to decrypt the messages using the above components. It is possible to produce the function **E**=**D**, and such functions uses the total key length as strength. Though 64 bits may offer a very good security, a 128 bit or 256 bit will definetely use for any mission critical applications. It also aids in improving security because of the increase in the key space, which makes it less prone to brute-force key search and other kind of attacks like plaintext attack, choosed plaintext attack, differential plaintext attack etc. The Stream ciphers are the algorithms that uses the function to encrypt the plaintext in a stream more or less like reading a file character by character and encoding it. The block ciphers are used to encrypt the plaintext in a pre-defined size of blocks. Say 32bytes, 64 bytes or 128 bytes. It is done to achieve speed.

Popular block ciphers are ***DES***, ***Blowfish*** etc, and stream ciphers are ***vernam ciphers*** or also known as ***one time pad (Considered to be very secure but practically almost impossible, aka OTP)*** ciphers.

In ***block Ciphers*** mode when you encode the message, the message size was not altered. Whereas in the ***stream cipher***, the message length gets altered. Thus these two are the major classification of the symmetric key cryptography. Various algorithms like DES,Blowfish come under the block cipher category.

The various attack methods used in the cryptanalysis against symmetric key cryptography are differential cryptanalysis, linear cryptanalysis and algebraic attacks.

Here is a comparison between the ***Symmetric key*** algorithms and ***Public key*** algorithms based on their key sizes.

Symmetric-Key Bit Length	Public-Key Bit Length
56 bits	384 bits
64 bits	512 bits
80 bits	768 bits
112 bits	1,792 bits
128 bits	2,304 bits

Public-key cryptography - as the name indicates, uses two kinds of functions and two different keys. The keys are terms as the ***private key*** and ***public key***. The *public key* is the one which is kept *'visible',* (i.e.), commonly transmitted over networks, etc. and the other one which is kept *'secret'* the private key, which is never revealed to anybody. Thus the system, uses both keys, it's commonly know as *'public/secret pair algorithms'* or *'private/public key algorithms'* though the name *public key cryptography* exists.

The functions used in the public key cryptography are like the following.
Consider the function **E,**

$$E_K(P) = C$$

which uses the **K**, the public key to encrypt plaintext **(P)** to produce ciphertext **(C)** can only be decrypted using another function **D,**

$$D_S(C) = P$$

Which uses **S**, which is the private key.

In some, (but not all) public/private algorithms, you can also use **S** to encrypt the plaintext that can be later decrypted using **K**, which is in use by some programs. The major advantage of the public/private pair cryptography is that is solves the issue of key distribution. But since, the functioning is concerned, the pair of functions deployed makes it less secure and there exists a list of attacks against the system. So, even the key strength of 768 bits pose a question threat to information security. Normally a key strength of 1024 and above is secure. But there are programs, which give about 4096 bits of key size which is an uncommon, but providing with a very good security. Such programs are often controlled by the U.S. govt. due to export restrictions to be used outside U.S.

Public-key algorithms are slower compared to the speed of symmetric key algorithms or secret-key algorithms. Also, public-key algorithms are prone to attack than the secret-key algorithms. Another main use of public-key algorithms is to produce ***digital signatures***, which plays the lead role in identifying the origin of the message. The *digital signature* thus provides the way to authenticate which is known as message non-repudiation as explained before. Public-key cryptography may be vulnerable to impersonation, but it is the sole responsibility of the user to protect his/her private key securely.

Digital signatures can be used to identify the message source and the integrity of the message using the digital fingerprint like conventional fingerprint matching. Slightest modification in the message will cause the fingerprinting mechanisms to fail and the whole message is discarded or reported back to the sender about the incident. It also provides the authentication mechanisms that requires users to sign on to a public terminal or access a network resource, by using the various implementations of the public-key cryptography. One such implementation is ***kerberos***^{tm}, which was developed at **MIT**, can be used for a network wide authentication procedure for a list from pre-qualified users using their digital signatures.

The real power can be brought out by using the combination of secret-key and public-key cryptosystems. Like using a secret-key algorithm to encrypt the data/file/message and then signing it with the public-key algorithm can be the highest possible security that can be given to a data/file/message. Also compressing the data/file/message with/without encryption (which uses some patented symmetric key algorithms!) will be more secure because the integrity is maintained to the lowest possible level.

The other mechanism, which is used along with above methods is called as ***hashing or message digesting.*** It means to produce a very small value, say $\mathbf{H_1}$ from a function **M** for the plaintext $\mathbf{P_1}$. It should be hard to find any other plaintext $\mathbf{P_2}$ **,** which satisfies $\mathbf{M(P_2)=H_2}$, which gives $\mathbf{H_1=H_2}$ proves that the plaintext **P1=P2**.
Example, function M,

$\mathbf{M(P_X)=H_X}$, Where **X** stands for the different plaintexts, that produces the hashed values. The function M is chosen, such that it is impossible to reverese the function to produce the plaintext from the hashed value (also called as *'hash'* or *'checksum'*).The common use of the hashing is to check the validity of the password. The hash thus produced should be greater that 128 bits to prevent a malicious user to break it. The only know attack against hash is the brute-force attack which is almost impossible because of the combination of the plaintext (actually the password) chosen. So by choosing a complicated password with the mixed symbols and alphanumerice characters, adds strength to the hashing algorithm by increasing the available keyspace to search for the key. The commonly used algorithms are ***MD4, MD5, SHA*** and ***SHA1***. As ***MD*** stands for ***Message Digest*** and ***SHA*** stands for ***Secure Hash Algorithm***.

In unix systems, a hashing algorithm called **MD4** was used to generate the checksum or the hash and nowadays, it is replaced with a more stronger algorithm called **MD5** which was a derivative from **MD4** after fixing an unpublished flaw in one of the hashing rounds.

Latest cryptographic Advancements.

Due to the modern methods of attacks and advancement in the computational speeds, have taken over the well known secure secret-key algorithm, ***DES***, ***D****igital* ***E****ncryption* ***S****tandard,* which dominated the world of cryptography for decades. A 40 bit version of the DES algorithm can be cracked by any available modern computer and a 56 bit can be broken by any cheaply available Super Computer or by a simple beo-wulf cluster. That pose a security threat and people have developed, developing various other algorithms like ***3DES*** also known as ***Triple DES*** etc,. The ***NIST*** *and* ***NSA*** plays an important role in deciding the algorithms. They analyse and recommend the algorithm for usage.

The latest trends have opened for the ***AES Advanced Encryptions Standards*** and many algorithms have been submitted for the review. Some of them approved and under analysis. They are IDEA, RC5, RC6 Rjindael, MARS, Two-fish, Blowfish, Diamond, Sapphire etc.

The inventions grow as the need increases. As per the truth, the importance of cryptography as given has a big response from scientific and legal bodies. Today, by the invention of high speed machines and high bandwidth transmission rates, people are planning to go further. The famous $\boldsymbol{e=mc^2}$ has stepped into cryptography also. The japanese have undertaken the improvement and implementation of ***quantum cryptography*** which uses the properties of the atoms and its inner particles for data integrity, which off course has a long way to go. The first commercially available quantum-cryptographic product is expected to be release around 2040's.

In the future versions of this draft, we will go in more details about platform and operating system specific measures that can be carried to protect, various products that are available for each platoforms. More on snake-oil algorithms, choosing of secure products, various advanced cryptoanalytic methods, and details about ***steganography*** - the other facet of cryptography which uses the commonly available formats like gifs, jpegs, wavs, bmps file formats to hide secret data and retrieve them without affecting the quality of the used format.

More to come… Contact goldie@checksum.org