A Comparative Analysis on Encryption and Decryption Algorithms

V. Magesh Babu, T. Shankar Ganesh, K. Ramraj

Bachelor of Computer Applications, Dept. Computer Applications and Software Systems, Sri Krishna Arts and Science College, Coimbatore – 641 008.

Abstract- In this modern world of communications, cryptography has an important role in the security of data transmission and is the best method of data protection against passive and active fraud. Cryptography is an algorithmic process of converting a plain text or clear text message to a cipher text or cipher message based on an algorithm that both the sender and receiver know. There are a number of algorithms for performing encryption and decryption, but comparatively few such algorithms have stood the test of time. The most successful algorithms use a key. In this paper, we may gain knowledge about the cryptography algorithms and its role in Encryption and Decryption.

Index Terms- cipher text, decryption, encryption, key generation

I. Introduction

Encryption is a process of coding information which could either be a file or mail message in into cipher text a form unreadable without a decoding key in order to prevent anyone except the intended recipient from reading that data. Decryption is the reverse process of converting encoded data to its original un-encoded form, plaintext. A key in cryptography is a long sequence of bits used by encryption/ decryption algorithms. For example, the following represents a hypothetical 40-bit key:
00001010 01101001 10011110 00011100 01010101

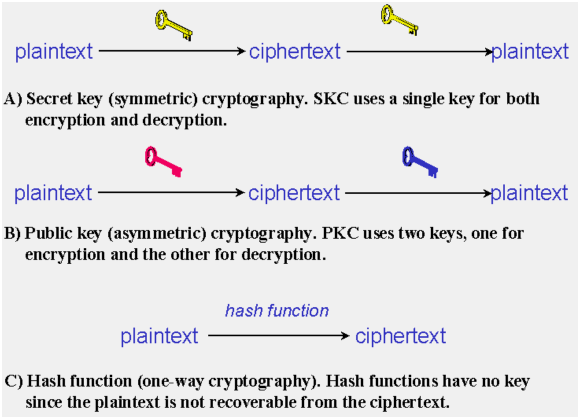
A given encryption algorithm takes the original message, and a key, and alters the original message mathematically based on the key's bits to create a new encrypted message. Likewise, a decryption algorithm takes an encrypted message and restores it to its original form using one or more keys. To encode plaintext, an encryption key is used to impose an encryption algorithm onto the data. To decode cipher, a user must possess the appropriate decryption key.

II. BACKGROUND OF ENCRYPTION AND DECRYPTION ALGORITHMS

CRYPTOGRAPHY is an algorithmic process of converting a plain text or clear text message to a cipher text or cipher message based on an algorithm that both the sender and receiver know, so that the cipher text message can be returned to its original, plain text form. In its cipher form, a message cannot be read by anyone but the intended receiver. The act of converting a plain text message to its cipher text form is called enciphering. Reversing that act is deciphering. There are two primary approaches to encryption: symmetric and public-key. Symmetric encryption is the most common type of encryption and uses the same key for encoding and decoding data. This key is known as a session key. Public-key encryption uses two different keys, a public key and a private key. One key encodes the message and the other decodes it. The public key is widely distributed while the private key is secret.

A. *Types of cryptographic algorithms*- There are several ways of classifying cryptographic algorithms. For purposes of this report they will be categorized based on the number of keys that are employed for encryption and decryption, and further defined by their application and use. The following are the three types of Algorithm that are discussed

1. Secret Key Cryptography (SKC): Uses a single key for both encryption and decryption
2. Public Key Cryptography (PKC): Uses one key for encryption and another for decryption
3. Hash Functions: Uses a mathematical transformation to irreversibly "encrypt" information



A) Secret key (symmetric) cryptography. SKC uses a single key for both encryption and decryption.

B) Public key (asymmetric) cryptography. PKC uses two keys, one for encryption and the other for decryption.

C) Hash function (one-way cryptography). Hash functions have no key since the plaintext is not recoverable from the ciphertext.

Figure 1: ijsrp.org Three types of cryptography: secret-key, public key, and hash function

Symmetric Key Cryptography- The most widely used symmetric key cryptographic method is the Data Encryption Standard (DES), published in 1977 by the National Bureau of Standards. DES It is still the most widely used symmetric-key approach. It uses a fixed length, 56-bit key and an efficient algorithm to quickly encrypt and decrypt messages. It can be easily implemented in hardware, making the encryption and decryption process even faster. In general, increasing the key size makes the system more secure. A variation of DES, called Triple-DES or DES-EDE (encrypt-decrypt-encrypt), uses three applications of DES and two independent DES keys to produce an effective key length of 168 bits [ANSI 85].

Public/Private Key Cryptography-Asymmetric key cryptography overcomes the key management problem by using different encryption and decryption key pairs. Having knowledge of one key, say the encryption key, is not sufficient enough to determine the other key - the decryption key. Therefore, the encryption key can be made public, provided the decryption key is held only by the party wishing to receive encrypted messages (hence the name public/private key cryptography). Anyone can use the public key to encrypt a message, but only the recipient can decrypt it.

Hash functions-"Is a type of one-way function this are fundamental for much of cryptography. A one way function - is a function that is easy to calculate but hard to invert. It is difficult to calculate the input to the function given its output. The precise meanings of "easy" and "hard" can be specified mathematically. With rare exceptions, almost the entire field of public key cryptography rests on the existence of one-way functions.

III. STUDIES AND FINDINGS

A. *Secret/Symmetric key cryptography*

Secret key cryptography schemes are generally categorized as being either stream ciphers or block ciphers. Stream ciphers operate on a single bit (byte or computer word) at a time, and implement some form of feedback mechanism so that the key is constantly changing.

❖ Encryption algorithm

Step 1: Generate the ASCII value of the letter

Step 2: Generate the corresponding binary value of it. [Binary value should be 8 digits e.g. for decimal 32 binary number should be 00100000]

Step 3: Reverse the 8 digit's binary number

Step 4: Take a 4 digits divisor (>=1000) as the Key

Step 5: Divide the reversed number with the divisor

Step 6: Store the remainder in first 3 digits & quotient in next 5 digits (remainder and quotient wouldn't be more than 3 digits and 5 digits long respectively. If any of these are less than 3 and 5 digits respectively we need to add required number of 0s (zeros) in the left hand side. So, this would be the cipher text i.e. encrypted text.

Now store the remainder in first 3 digits & quotient in next 5 digits.

❖ Decryption algorithm

Step 1: Multiply last 5 digits of the cipher text by the Key

Step 2: Add first 3 digits of the cipher text with the result produced in the previous step

Step 3: If the result produced in the previous step i.e. step 2 is not an 8-bit number we need to make it an 8- bit number

Step 4: Reverse the number to get the original text i.e. the plain text.

Example

Let, the character is "T". Now according to the steps we will get the following:

Step 1: ASCII of "T" is 84 in decimal.

Step 2: The Binary value of 84 is 1010100. Since it is not an 8 bit binary numbers we need to make it 8 bit number as per the encryption algorithm. So it would be 01010100

[1] 0	[2] 1	[3] 0	[4] 1	[5] 0	[6] 1	[7] 0	[8] 0

Step 3: Reverse of this binary number would be 00101010

[9] 0	[10] 0	[11] 1	[12] 0	[13] 1	[14] 0	[15] 1	[16] 0

Step 4: Let 1000 as divisor i.e. Key

Step 5: Divide 00101010 (dividend) by 1000(divisor)

Step 6: The remainder would be 10 and the quotient would be 101.

So as per the algorithm the cipher text would be 01000101 which is
ASCII 69 in decimal i.e. "E"

[17] 0	[18] 1	[19] 0	[20] 0	[21] 0	[22] 1	[23] 0	[24] 1

B. *Public key cryptography*

RSA is one of the first practicable public-key cryptosystems and is widely used for secure data transmission. In such a cryptosystem, the encryption key is public and differs from the decryption key which is kept secret. In RSA, this asymmetry is based on the practical difficulty of factoring the product of two large prime numbers, the factoring problem. RSA stands for Ron Rivets, Adi Shamir and Leonard Adleman, who first publicly described the algorithm in 1977. Clifford Cocks, an English mathematician, had developed an equivalent system in 1973, but it wasn't declassified until 1997.

A user of RSA creates and then publishes a public key based on the two large prime numbers, along with an auxiliary value. The prime numbers must be kept secret. Anyone can use the public key to encrypt a message, but with currently published methods, if the public key is large enough, only someone with knowledge of the prime factors can feasibly decode the message. Breaking RSA encryption is known as the RSA problem. It is an open question whether it is as hard as the factoring problem.

❖ Key generation

RSA involves a *public key* and a *private key*. The public key can be known by everyone and is used for encrypting messages. Messages encrypted with the public key can only be decrypted in a reasonable amount of time using the private key. The keys for the RSA algorithm are generated the following way:

1. Choose two distinct prime numbers p and q.
   - For security purposes, the integer's p and q should be chosen at random, and should be of similar bit-length. Prime integers can be efficiently found using a primarily test.
2. Compute $n = pq$.
   - n is used as the modulus for both the public and private keys. Its length, usually expressed in bits, is the key length.
3. Compute $\varphi(n) = \varphi(p)\varphi(q) = (p - 1)(q - 1) = n - (p + q - 1)$, where φ is Euler's totient function.

4. Choose an integer *e* such that $1 < e < \varphi(n)$ and gcd(e, $\varphi(n)$) = 1; i.e., *e* and $\varphi(n)$ are coprime.
    - *e* is released as the public key exponent.
    - *e* having a short bit-length and small Hamming weight results in more efficient encryption – most commonly $2^{16} + 1 = 65{,}537$. However, much smaller values of *e* (such as 3) have been shown to be less secure in some settings.[5]
5. Determine *d* as $d \equiv e^{-1}$ (mod $\varphi(n)$); i.e., *d* is the multiplicative inverse of *e* (modulo $\varphi(n)$).

- This is more clearly stated as: solve for *d* given $d \cdot e \equiv 1$ (mod $\varphi(n)$)
- This is often computed using the extended Euclidean algorithm. Using the pseudo code in the *Modular integers* section, inputs *a* and *n* correspond to *e* and $\varphi(n)$, respectively.
- *d* is kept as the private key exponent.

The *public key* consists of the modulus *n* and the public (or encryption) exponent *e*. The *private key* consists of the modulus *n* and the private (or decryption) exponent *d*, which must be kept secret. *p*, *q*, and $\varphi(n)$ must also be kept secret because they can be used to calculate *d*.

- An alternative, used by PKCS#1, is to choose *d* matching $de \equiv 1$ (mod λ) with

$\lambda = \operatorname{lcm}(p - 1, q - 1)$, where lcm is the least common multiple. Using λ instead of $\varphi(n)$ allows more choices for *d*. λ can also be defined using the Carmichael function, $\lambda(n)$.

- The ANSI X9.31 standard prescribes, IEEE 1363 describes, and PKCS#1 allows, that *p* and *q* match additional requirements: being strong primes, and being different enough that Fermat factorization fails.

❖ Encryption

Alice transmits her public key (n, e) to Bob and keeps the private key *d* secret. Bob then wishes to send message *M* to Alice.

He first turns *M* into an integer *m*, such that $0 \leq m < n$ by using an agreed-upon reversible protocol known as a padding scheme. He then computes the cipher text *c* corresponding to

$$c \equiv m^e \pmod{n}$$

This can be done quickly using the method of exponentiation by squaring. Bob then transmits *c* to Alice.

Note that at least nine values of *m* will yield a cipher text *c* equal to *m*,[note 1] but this is very unlikely to occur in practice.

❖ Decryption

Alice can recover *m* from *c* by using her private key exponent *d* via computing

$$m \equiv c^d \pmod{n}$$

Given *m*, she can recover the original message *M* by reversing the padding scheme.

C. *Cryptographic hash function*

Hash functions (a type of one-way function) are fundamental for much of cryptography. In this application, functions are characterized and evaluated in terms of their ability to withstand attack by an adversary. More specifically, given a message x, if it is computationally infeasible to find a message y not equal to x such that H(x) = H(y) then H is said to be a weakly collision-free hash function. A *strongly collision-free hash function* H is one for which it is computationally infeasible to find any two messages x and y such that H(x) = H(y).

The requirements for a good cryptographic hash function are stronger than those in many other applications (error correction and audio identification *not* included). For this reason, cryptographic hash functions make good stock hash functions--even functions whose cryptographic security is compromised, such as MD5 and SHA-1. The SHA-2 algorithm, however, has no known compromises". Hash function can also be referred to as a function with certain additional security properties to make it suitable for use as a primitive in various information security applications, such as authentication and message integrity. It takes a long string (or message) of any length as input and produces a fixed length string as output, sometimes termed **a** message digest or a digital fingerprint.

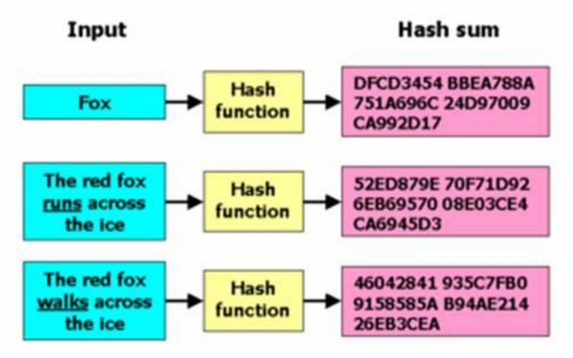

Input
Hash sum
Fox
Hash function
DFCD3454 BBEA788A 751A696C 24D97009 CA992D17
The red fox runs across the ice
Hash function
52ED879E 70F71D92 6EB69570 08E03CE4 CA6945D3
The red fox walks across the ice
Hash function
46042841 935C7FB0 9158585A B94AE214 26EB3CEA

Figure 2: ijsrp.org Hash function

Above figure illustrates the proper and intended use of public/private key cryptography for sending confidential messages. In the illustration, a user, Bob, has a public/private key pair. The public portion of that key pair is placed in the public domain (for example in a Web server). The private portion is guarded in a private domain, for example, on a digital key card or in a password-protected file.

For Alice to send a secret message to Bob, the following process needs to be followed:

1. Alice passes the secret message and Bob's public key to the appropriate encryption algorithm to construct the encrypted message.
2. Alice transmits the encrypted message (perhaps via e-mail) to Bob.
3. Bob decrypts the transmitted, encrypted message with his private key and the appropriate decryption algorithm.

Bob can be assured that Alice's encrypted secret message was not seen by anyone else since only his private key is capable of decrypting the message

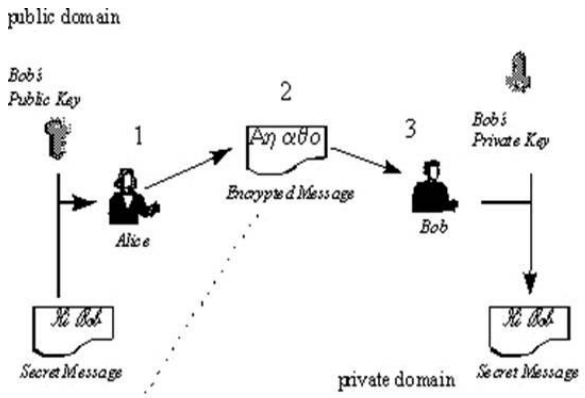


Figure 3: ijsrp.org Communication between alias and bob in PGP e-mail encryption program.

Both Are Used Together - Secret key and public key systems are often used together, such as the AES secret key and the RSA public key. The secret key method provides the fastest decryption, and the public key method provides a convenient way to transmit the secret key. This is called a "digital envelope." For example, the PGP e-mail encryption program uses one of several public key methods to send the secret key along with the message that has been encrypted with that secret key.

Get Faster - Get Stronger - It has been said that any encryption code can be broken given enough time to compute all permutations. However, if it takes months to break a code, the war could already be lost, or the thief could have long absconded with the money from the forged financial transaction.

D. Use in software

In UNIX systems things are difficult if you are user since super user has full access your keys/log keystrokes. Securing UNIX is very large topic, there is a wealth of information available however, but it revolves around the same principles of any operating systems.

- Keep software up-to-date, this is how most remote attacks succeed.
- Use an anti-virus scanner, and keep it update.
- Don't run mystery attachments, especially executable once are data formats.
- Run a little software as possible if u don't need something to get your job done, remove it, it won't be a problem that way.
- Keep the machine physically secure if possible use BIOS passwords and so on to secure access to the machine.

IV. CRYPTOGRAPHY REVIEW

Cryptography is used for four basic purposes:

- Confidentiality/privacy
- Integrity
- Authentication
- Nonrepudiation

Confidentiality/privacy: Ensuring the data can't be revealed to un-authorized entities. This involves full encryption of user data. Those who can't decrypt the message see only gibberish.

Integrity: Ensuring that data has not been modified or corrupted. It is typically verified using cryptographic data checksums, which is a less expensive operation than full encryptions of data. The data isn't secret-anyone can see/read it, but it can't be modified without detection.

Authentication- Securely proving entities are who they claim are, so that they may trust each other.

Nonrepudiation: Preventing an entity who took part in a communication from later denying or part of that communication.

V. CONCLUSION

Cryptography is used to achieve few goals like Confidentiality, Data integrity, Authentication etc. of the send data. Now, in order to achieve these goals various cryptographic algorithms are developed by various people. For a very minimal amount of data those algorithms wouldn't be cost effective since those are not designed for small amount of data. A single algorithm is used for both encryption and decryption i.e. it is fallen under secret key cryptographic algorithm. But as public key cryptography is more secured then secret key cryptography our next task would be to develop and design a public key cryptographic algorithm in a simple manner as it is done in this paper.

REFERENCES

[1] https://www.google.co.in/webhp?sourceid=chrome-instant&ion=1&espv=2&ie=UTF-8#q=encryption+and+decryption+algorithm&spell=1

[2] http://en.wikipedia.org/wiki/RSA_(cryptosystem)

[3] http://en.wikipedia.org/wiki/Encryption

[4] http://homepages.uel.ac.uk/u0430614/Encryption%20index.htm

[5] http://msdn.microsoft.com/en-

[6] us/library/e970bs09(v=vs.110).aspx

[7] http://msdn.microsoft.com/en-us/library/ee497974.aspx

[8] http://www.obviex.com/samples/encryption.aspx

[9] http://www.networksorcery.com/enp/data/encryption.htm

[10] http://www.cryptoforge.com/security.htm

[11] http://www.eecs.berkeley.edu/~messer/netappc/Supplements/13-encalgs.pdf

AUTHORS

First Author- V. Magesh Babu, Bachelor of Computer Applications, Sri Krishna Arts and Science College,mageshbabuv13bca128@skasc.ac.in.

Second Author- T. Shankar Ganesh, Bachelor of Computer Applications, Sri Krishna Arts and ScienceCollege,shankarganesht13bca144@skasc.ac.in.

Third Author- K. Ramraj, Bachelor of Computer Applications, Sri Krishna Arts and Science College,ramrajk13bca138@skasc.ac.in.