An Effective Public Key Cryptography Based on Factoring and Discrete Logarithm

Dr. Khader M. Titi [1]
Dr. Osama Marie [2]
[1]Irbid National University
e-mail:drkhmt@gmail.com
[2]Al-Quds Open University

Abstract

An effective and new scheme for public key cryptosystem is proposed. The system employs computations in Z_n^* where n is a composite modulus of three large prime numbers. There is similarity with the most widely used public key encryption scheme the RSA scheme regarding the mathematical background. Regarding calculation needs the new approach is unchangeable and it more efficient compare with the best known scheme, when measured with both public and private calculation.

Introduction

Data communication is an important part of our living. Therefore, protection of data from misuse is essential. A cryptosystem defines a pair of data transformations called encryption and decryption. Encryption is applied to the plain text which is the data to be communicated to produce cipher text which is the encrypted data using encryption key. Decryption uses the decryption key to convert cipher text to plain text or the original data.

With strong encryption, computer users can send confidential contracts or data by e-mail, or safely store corporate strategy on a notebook, or carry home spreadsheets on a floppy disk. The encryption software may even be free. To improve the protection mechanism Public Key Cryptosystem was introduced in 1976 by Whitfield Diffe and Martin Hellman of Stanford University [[i]]. It uses a pair of related keys one for encryption and other for decryption. One key, which is called the private key, is kept secret and other one known as public key is disclosed to the public. The message is encrypted with public key and can only be decrypted by using the private key. So, the encrypted message cannot be decrypted by anyone who knows the public key and thus secure communication is possible. RSA [[ii]] is the most popular public key algorithm. In relies on the factorization problem of mathematics that indicates that given a very large number it is quite impossible in today’s aspect to find two prime numbers whose product is the given number. As we increase the number the possibility for factoring the number decreases. Thus, we need very large numbers for a good Public Key Cryptosystem.

Authentication, confidentiality and data integrity can be addressed by studying cryptographic techniques. In using such techniques, it is predictable that information in transmit through the Internet can bypass through various computers before it arrives its target. A malicious user of any of the intermediary computers can monitor the Internet traffic, eavesdrop, intercept, change or replace the data through its entire path. Cryptographic techniques can be used to protect these data. Encryption is the process that makes information indecipherable (cipher text) unless having a decryption key. It uses mathematical algorithms and processes to convert intelligible plain text to unintelligible cipher text and vice versa [[iii]]. It can, therefore, reduce risks from an eavesdropping on a network.

Therefore, there are two types of key encryption systems: symmetric and asymmetric systems. Asymmetric Key Coding System is commonly called as the Public Key Coding System (PKCS). The public and private keys are mutual in the RSA algorithm. Thus, if one is used to encrypt, the other can be used to decrypt. In this method it is assumed that the individual is the only one capable of encryption the message with his/her own private key, but anyone with the mutual public key may decrypt the message. If the message has been published or broadcast, anyone receiving it can be satisfied that the first party performed the encryption and no one else. This is similar to a signature on a document, and has become known as a "Digital Signature (DS)". Digital signatures can be used to provide authenticity.

Conventional encryption is still be used because it is more efficient. The best asymmetric methods are much slower than strong symmetric methods. This is why PGP [iv] uses both. When it is used for privacy, the PGP program chooses a 128 bits random number, called a "session key" and uses that as the key to encrypt the bulk of the message. It then encrypts the relatively small session key with each of the public key of the intended recipient and adds it to the encrypted data. The recipient, on receipt of the message, can then use their private key to decrypt the session key and use that to decrypt the main information.

Thus, we may assume that cryptography is one of the most important tools that enable networks and Internet applications because cryptography makes it possible to protect electronic information. The effectiveness of this protection depends on a variety of mostly unrelated issues such as cryptographic key size, protocol design, and password selection. Each of these issues is equally important: if a key is too small, or if a protocol is badly designed or incorrectly used, or if a password is poorly selected or protected, then the protection fails and improper access can be gained.

RSA (Public-Key Cryptosystem)

The RSA cryptosystem is a public-key cryptosystem that presents both encryption and digital signatures (authentication). Ronald Rivest, Adi Shamir, and Leonard Adleman developed the RSA system in 1977 [v]. RSA stands for the first letter in each of its discoverers' last names.

In public key cryptography, two keys are used; a public key and a private key. Usually the public key is used for encrypting messages and is placed in a public register or a file while the private key is used for decrypting messages and is kept secret. For digital signatures, however, a document is signed with a private key and is verified with the corresponding public key [vi]. Therefore, in a public key cryptosystem, it is important that the private key is kept safely. Because of the large size of a cryptographically-strong key, it is not feasible for a user to remember the private key and enter each time it is required. Instead, the private key is usually encrypted with a user chosen password and is stored in some medium like the hard disk, smart card etc. To retrieve the key, the user will have to enter the same password in order to perform successful decryption of the key. This therefore, brings the security of the whole cryptosystem to rely on the strength of the password. A common problem with a password-based method is the low entropy in user chosen passwords, which may be exploited by an attacker to launch password guessing attacks. The space of passwords is very limited. For an eight-character password, there are approximately 2×1014 combinations of English characters, both upper and lower case and digits. This is very small as compared to the size of a 1,024-bit RSA key [vii].

Digital Signatures

Digital signature is an electronic signature that can be used to authenticate the identity of the sender of a message or the signer of a document, and maybe to ensure that the original content of the message or document after being sent is unaffected. Digital signature has been used since 1976, when Whifield Diffie and Martin Hellman introduced the digital signature as an application of public key cryptography [viii]. Only recently have businesses and governments started to use digital signature technology to protect sensitive documents on the Internet.

All digital signature technologies employ a public key infrastructure (PKI). Under PKI, an individual has a pair of keys: a private key and a public key. A digital signature is obtained as the sender signs a document with his private key. When the recipients receive the signed document, they use the sender’s public key to authenticate the document and verify that it has not been tampered with in transit.

A digital signature is an identifier, which could be used to authenticate the sender of an electronic message or document. This approach could also be used to guarantee the integrity of the message or document that no modifications have been made since it was signed as well as to date/time-stamp the document at signing. Additionally, the signatory person cannot easily repudiate or refuse to acknowledge his/her digital signature, nor can the document be easily faked.

The digital signature uses cryptographic tools to create an electronic identifier, however it can be used with any message, whether the message is encrypted or not. Therefore, digital signatures can go with an unencrypted or an encrypted message. A user creates a digital signature with a private key that he keeps to himself. He then attaches this signature to a document and sends it to others. His/her private key is mathematically linked to a public key that he posts on a public key server. He then tells the recipient where his/her public key is stored. The recipient can then retrieve the sender's public key and reverse the process to determine the authenticity of the document. The integrity of a digital signature can be compromised if someone gains illegal access to the computer that runs the encryption software.

In common, there are four common reasons for applying a digital signature to communications:

1. *Authentication* – ensures that a principal is really who he or she claims to be.
2. *Data authentication* – ensures that the data origin cannot be forged.
3. *Integrity* –ensure that the data have not been modified by unauthorized person. Both communicated parties will always wish to be sure that a message has not been altered during transmission. Encryption of the message makes it difficult for a third party to read it, but that third party may still be able to modify it, possibly maliciously
4. *Non-repudiation* - In a cryptographic context, the word repudiation refers to the act of disclaiming responsibility for a message (that is, claiming it was sent by a third party). A message's recipient could insist the sender attach a signature in order to prevent later repudiation, since the recipient may show the message to a third party to reinforce a claim as to its origin. Loss of control of a user's private key will mean that all digitally signatures using that key become suspect.

Key Management System

Key management system deals with the secure generation, distribution, and storage of keys. Secure techniques of key management are very significant. When a key is randomly generated it has to stay secret to prevent unfortunate accident. In practice, most attacks on public-key systems will most likely be aimed at the key management level, rather than at the cryptographic algorithm itself.

According to data encryption, a control of the key that is used for decryption is essential. The key must be managed securely so the unauthorized individuals cannot access to it. According to [ix], the key manage lifecycle is documented as follows:

- User registration
- System and user initialization
- Keying material installation
- Key establishment
- Key registration
- Operational use
- Storage of keying material
- Key update
- Key recovery
- Key registration and destruction
- Key revocation

Users must be able to securely obtain a key pair suited to their efficiency and security needs. There must be a way to look up other people's public keys and to publicize one's own public key. Users must be capable to legally get others' public keys; or else, an intruder can either change public keys listed in a directory, or pretend to be another user. Certificates are used for this purpose. Certificates must not be faked by any one. The issuer of certificates must proceed in a secure method. In particular, the issuer must authenticate the identity and the public key of an individual before issuing a certificate to that individual. If someone's private key is lost or compromised, others must be made aware of this, and so they will no longer encrypt messages under the invalid public key nor accept messages signed with the invalid private key. Users must be able to store their private keys securely, so no intruder can obtain them, yet the keys must be readily accessible for legitimate use. Keys need to be valid only until a specified expiration data but the expiration date must be chosen properly and publicized in an authenticated channel.

Key Length

The RSA cryptosystem is rapidly loosing its attractiveness. This is mainly due to the enormous key lengths necessary to make RSA secure. A report by Lenstra and Verheul [x], several organizations started to increase the key size of an RSA modulus up to 2048 bits. On the other hand, computer hardware ongoing development and enhanced programmable devices such as smart card cryptographic processors are becoming more and more efficient over time,

The advantage of forcing an adversary to fall back on a brute-force attack to crack a code is that the key length can be used as a quantitative measure for comparison with other algorithms. For example, a 32 bits key would only take seconds to search, on any advance computer, regardless of which of the common symmetric ciphers was deployed. On the other hand, a 2048 bits key would provide secure encryption for a number of years to come, even when future advances in computer technology are taken into account. When taking into consideration a specific encryption algorithm, a decision has to be made about the value of the original data and the time period through which it must be reserved secure. The length of the key will provide the means to evaluate these matters.

Secure Generation Random Numbers for Keys

Random numbers occupy a very important role in the security of any cryptographic system. In any cryptographic system at a particular point a secret random key has to be generated and if it is at all likely to estimate the key then the power of the cipher has been assured.

Generation of random numbers to be used as a key for use in cryptography is one of the most difficult security issues to be surly employed and accomplish rightly. Several methods and techniques have been employed to generate random numbers including using specialized hardware and software. The major cause lies in developing a random number system which is cross-platform compatible (operating system, hardware, and network). Different architectures have very few hardware devices in common. Those that are common to nearly all platforms provide random numbers fairly slowly.

whether using a secret-key cryptosystem or a public-key cryptosystem, it is necessary to have a way to get random numbers for key generation. The main features of a good source are that it produces numbers that are unknown and unpredictable by potential adversaries. Random numbers obtained from a physical process are in principle the best, since many physical processes appear truly random.

One of the efficient random-numbers generating algorithms is called Portable Random Number Generated [xi]. This approach has great advantages in generating random numbers with full period and high quality random numbers generation. The approach uses three variables X, Y, and Z to generate random numbers. The algorithm of this approach is as follows:

- **Select a suitable seed Xi**
- **Calculate Xi+1**=171 * (Xi MOD 177)) – (2*X/177))
    - If X<0 then
        - X=X+30269
- **Calculate Yi+1**= 172 * (Yi MOD 176)) – (35*Y/176))
    - If Y<0 then
        - Y=Y+30307
- **Calculate Zi+1**= (170 *(Zi MOD 178)) – (63*(Z/178))
    - If Z<0 then
        - Z=Z+30323
- **Temp**=(X/30269)+(Y/30307) + (Z/30323)
- **Random** = Temp – Int (temp)

In RSA cryptosystem if the random number generated with this method is not a prime number it will be ignored.

If a secure RNG is used to generate a cryptographic key, the length of the seed shall be more or equal to the length of the cryptographic key being generated. For example, it does not make sense to generate an RSA 256 bit key using a random generator which has the seed length of 112 bits. Otherwise, one would be able to find the RSA key by trying 2 to the power of 112 seeds, which would effectively decrease the key strength from 256 to 112 bits.

The Factorization Problem

The factorization problem is how to find two large numbers p and q given a composite number n that is the product of the two numbers p and q. While finding large prime numbers is a relatively easy task, the problem of factoring the product of two such numbers is considered to be computationally difficult if the primes are carefully selected. Rivest, Shamir and Adleman [xxxxxx] developed the RSA public-key cryptosystem based on the difficulty of this problem. While the factorization problem has received some attention over many years from many mathematicians, it is only in the past 20 years that significant progress has been made towards its resolution. Since, the invention of the RSA cryptosystem in 1978 inspired many mathematicians to study the problem. Additionally, high-speed computers became available for the implementation and testing of sophisticated algorithms.

The RSA Problem is now more than a quarter century old [xii]. The robust simplicity of the problem has led to several observations over the years, some yielding attacks, others avoiding them. Digital signature and Public-key encryption schemes have been developed whose power is derived from the RSA Problem. The question now is how much the security of the RSA Problem depends on factoring, and as with any hard problem in cryptography, whether any methods more robust than those currently available for solving the problem will ever be found.

In spite of the widely used of RSA schema in the whole world, RSA cryptosystem is rapidly beginning to lose its magnetism. This is mainly due to the huge key lengths necessary to let RSA be more and more secure. Recently, [9] several organizations suggested to increase the key size of an RSA modulus up to 2048 bits. Nevertheless, information technologies enhanced programmable devices such as smart card cryptographic processors that become more and more robust and efficient over time. Thus, considerable research efforts in the field could be saved by the simple fact of allowing several prime factors to appear in the factorization of RSA modulus. For example, one can imagine to perform an RSA exponentiation with a 2048-bit modulus of the form

$$N = p.q.r.s$$

where p; q; r; s are 512-bit primes. Another possible choice is a modulus of the form

$$N = p^k.q^{k.m}$$

The security level of RSA still unchanged except when major scientific discoveries such as new factoring algorithms are carried out in the space of factorization. In fact, the current state of the process includes two main aspects of factorization algorithms. The first one presents a running time which depends on the total length of the number to be factored while the running time of algorithms belonging to the second aspect only depends on the length of the factors. The main threat for RSA comes from the first aspect of factorization algorithms.

In RSA public-key encryption, suppose Ali encrypts a plaintext M for Bakr using Bakr's public key (n, e) by computing the ciphertext

$$C = M^e \pmod n$$

where n, the modulus, is the product of two large prime numbers, and e, the public exponent, is an odd integer $e \geq 3$ that is relatively prime to $\Phi(n)$, the order of the multiplicative group Z^*_n.

Bakr, who knows the corresponding RSA private key (n, d), can easily decrypt, since

$$d.e = 1 \pmod{\Phi(n)} \text{ implies that}$$

$$M = C^d \pmod n$$

An opponent may learn C by eavesdropping, and may very well also know Bakr's public key; however such an opponent should not be able to compute the corresponding plaintext M. Thus, The RSA Problem may be formalized as follows:

Given an RSA public key (n, e) and a ciphertext $C = M^e \pmod n$, compute M.

To solve the RSA Problem an opponent, who doesn't know the private key, must invert the RSA function. The RSA Assumption is that the RSA Problem is hard to be solved when the modulus n is sufficiently large and randomly generated, and the plaintext M and the ciphertext C is a random integer between 0 and $n - 1$. The randomness of the plaintext M over the range $[0, n-1]$ is significant in the assumption. If M is known to be from a small space, then an opponent can solve for M by trying all possible values for M.

The RSA Problem is the basis for the security of RSA public-key encryption as well as RSA digital signature schemes. The RSA Problem is clearly no harder than integer factoring, since an opponent who can factor the modulus n can compute the private key (n, d) from the public key (n, e). However, it is not clear whether the converse is true, that is, whether an algorithm for integer factoring can be efficiently constructed from an algorithm for solving the RSA Problem.

The RSA problem is obviously not harder than integer factoring, as an adversary who can factor the modulus n can be able to compute the private key (d, n) from the public key (e, n). But, it is not obvious whether the opposite is true, that is, whether an algorithm for integer factoring can be efficiently constructed from an algorithm for solving the RSA Problem.

Low Public Exponent RSA

A user of the RSA cryptosystem might practically want to use a public exponent e that is relatively small: common choices are

$$e = 3 \text{ or } e = 2^{16} + 1 = 65537.$$

Using a small public exponent results in quicker public-key encryption and quicker public-key signature confirmation.

Does this process make the RSA weak and vulnerable?

If the public exponent is short and the plaintext M is very small, then the RSA procedure might be easy to invert: in particular,

$$\text{if} \quad M < \sqrt[e]{N} \quad , \text{ then}$$

$$C = M^{e}$$

over the integers, so M can be recovered as

$$M = \sqrt[e]{C}$$

The Proposed Public-Key Encryption Algorithm

A very significant modifications have been done on the original RSA public-key encryption algorithm. Thus, new public-key encryption algorithm has been designed, emerged and tested using computer software. The new new public-key encryption algorithm works as follows:

1- Select three prime numbers g ,f and r
2- Calculate n = g * f * r
3- Calculate Φ (n) = (g-1) * (f-1) * (r-1)
4- Find the residues of residence(RRs) of Φ (n) such that :
   - GCD(Φ(n), RRs)=1
5- Select **c** one of the residence of Φ (n) such that :
   - 1<c< Φ (n) and GCD (Φ (n), c) =1
6- Find **d** from residues of residence of Φ (n) such that :
   - (d * c) ≡1 mod Φ (n)
7- Calculate $\mathbf{M} = (\mathbf{C}^{\Phi(n)} \bmod n)^{d} * \mathbf{C} \bmod n$
8- Calculate $\mathbf{C} = (\mathbf{M}^{\Phi(n)} \bmod n)^{c} * \mathbf{M} \bmod n$
9- Public key ➔ K_{pub}= { c , n }
10- Private key ➔ K_{pri} = { d , n }

The following example will illustrate the mRSA scheme implemented using small prime numbers:

- Select g= 5 , f= 7 , r = 13
- n = g. f. r ➔ n = 5x7x13= 455
- Φ (n) = (g-1) * (f-1) * (r-1) = 4x6x12 = 288
  - ∴ Residues of residence = 5,7,11, 13, 15, 17,….., 287.
- Let c= 17
- Find d such that (d x c) mod Φ (n) =1
  - 7d mod Φ (n)=1 ➔ 17d mod 288 =1
  - ∴ d= 17 ,
  - Let M=7
- $\mathbf{C} = (\mathbf{M}^{\Phi(n)} \bmod n)^{c} * \mathbf{M} \bmod n$
  - $C = (7^{288} \bmod 455)^{17} * 7 \bmod 455$
  - C= 7
- $\mathbf{M} = (\mathbf{C}^{\Phi(n)} \bmod n)^{d} * \mathbf{C} \bmod n$
  - $M = (7^{288} \bmod 455)^{17} * 7 \bmod 455$
  - M= 7

The standard RSA and the new proposed scheme problem is obviously no harder than integer number factoring, since an opponent who can factor the modulus (n) can compute the private key (n, d) from the public key (n, c). However, with this new proposed schema n supposed to be quite large and composed of multiplying three numbers g, f and r. Their result means that the integer factoring problem for very small exponents could be easier than integer factoring of large one. De Laurentis [xiii] has shown that computing private key (n, d) from the corresponding encryption key (n, c) is as hard as factoring the modulus n into its prime factors. As already noted, given the factors g, f and r, it is easy to compute d from c, and conversely there is a probabilistic polynomial-time algorithm which takes as input n, c, and d, and which factors n into g, f and r.

If the modulus n was chosen as the product of three "sufficiently large" randomly-chosen prime numbers g , f and r, then the problem of factoring n appears to be very difficult. Thus, the private exponent d is protected from disclosure by the difficulty of factoring the modulus n.

This new propoaws scheme is seem to be more secure than the traditional RSA model. Hence, with this scheme the modulus

$$n = g * f * r$$

will be a very large number. Additionally, it will make it difficult for opponents to factorize n to find d.

In recent times, a report by Lenstra and Verheul [xiv] stated that, many organizations recommended to increase the key size of an RSA modulus up to 2048 bits. For instance, one can imagine to perform an RSA exponentiation with a 2048-bit modulus of the form N = pqr where p; q; r are 512-bit primes. Another possible choice is a modulus of the form $N = p^{kq}$ [75]. Therefore, the new proposed scheme provides such a proposal and consequently, the security level of RSA in such a case became fairly convenience.

The RSA cryptosystem [xxxxx] is still a de-facto standard in all branches of public key cryptography. However, it is rapidly losing its attractiveness. This is mainly due to the enormous key lengths necessary to make RSA secure. However, with this new proposed schema the encryption and decryption is much more secure due to the new formulas for calculating the plaintext and the ciphertext:

$$C = (M^{\Phi(n)} \bmod n)^{c} * M \bmod n$$
$$M = (C^{\Phi(n)} \bmod n)^{d} * C \bmod n$$

As we know in RSA schema

$$M = C^{d} \bmod n$$
$$C = M^{c} \bmod n$$

It is possible for the opponent to compute the value of M from a given C by the formula:

$$M = \sqrt[e]{C}$$

However, it is impossible to find M using the same way as above formula from the formula of new proposed scheme ;

$$C = (M^{\Phi(n)} \bmod n)^{c} * M \bmod n$$

This makes our new proposed scheme more reliable. In the new proposed scheme calculating C is more complex and need much time from the opponent to find the corresponding plain text M.

Algorithm Setup

The generation of the new proposed scheme keys is of paramount importance to the whole system. It is sure that the running time of the new proposed scheme algorithms only depends on the length of the factors. Hence, in the new proposed scheme three factors have been used g, f and r.

In order to use new proposed scheme some pre-calculations have to be made. The new proposed scheme is based on calculating powers of large integers modulo a large composite numbers. These numbers have to be generated before the system can be used. We want to create a public key *c* and a corresponding private key d. These keys are constructed as follows:

- Three large primes g , f , r are chosen using Portable Random Number Generating method,
- the modulus *n* is the product of the three primes *g, f and r.*

- *The public* key *c* is a randomly chosen number such that GCD (*c*, (*g*-1) * (*f*-1) * (r-1)) = 1 given that Φ(n) = (*g*-1) * (*f*-1) * (r-1) .
- the private key *d* is the multiplicative inverse of *c* modulus (*g*-1) * (*f*-1) * (r-1) that is: d*c mod Φ(n)=1

The public key *c* and the modulus *n* are made public whilst the rest of the numbers are kept secret. The public key c is selected according to several points:

- it should be large
- it should be prime
- must be selected using strong random number generation method. The method that had been used as a random number generation is the portable random number generation method

The New Proposed Scheme Encryption

A critical need for the proper functioning of the new proposed scheme algorithm is that the message M must be represented as an integer in the range [0, n-1], where n is the modulus. Our application converts text messages M to integers by using a simple mapping of each character to its ASCII (American Standard Code for Information Interchange) code. But encrypting only one character at a time is not only expensive in terms of the time required to encrypt and decrypt, but also in terms of security. This is because the encrypted integers would then only be from a small finite set (containing a maximum of as many integers as the number of ASCII characters). For new proposed scheme encryption scheme with the modulus size 1024 bits, about 100 characters can be encrypted at once. Lesser number of characters cause encryption and (especially) decryption to take significantly longer, whereas higher number of characters often violate the condition that the message M must lie in the interval [0, n-1].

The entire file to encrypt is processed as a group of strings each containing the specified number of characters (except possibly the last such string). Each character in such a string is converted to its 3- character wide ASCII code, and the entire resulting numeric string is our message M. Encrypting it is achieved by computing:

$$\mathbf{C} = (\mathbf{M}^{\Phi(n)} \mod n)^{c} * \mathbf{M} \mod n$$

The New Proposed Scheme Decryption

When the decryption routine is invoked, the function processes each encrypted integer. It does so by computing the value of

$$\mathbf{M} = (\mathbf{C}^{\Phi(n)} \mod n)^{d} * \mathbf{C} \mod n$$

By invoking mathematical function math power and stores the decrypted part in M. Of course, the values of d and n are read in beforehand. Here M however contains the integer representation of the message i.e. it is a string of numbers where each 3 character sequence signifies the ASCII code of a particular character. An inverse mapping to the relevant character is carried out and the message, now as was in the original file is displayed on the standard output. The decryption process is over once all the integers (ciphertext) in the encrypted file are processed in the described manner.

In our new proposed scheme public-key encryption, the entity student encrypts a plaintext M for the entity server or entity instructor using his/her private key (n, c) by computing the ciphertext:

$$\mathbf{C} = (\mathbf{M}^{\Phi(n)} \mod n)^{c} * \mathbf{M} \mod n \text{ ………………equation (1)}$$

where n, the modulus, is the product of three large primes, and c, the public exponent, is an odd integer c ≥ 3 that is relatively prime to Φ(n), the order of the multiplicative group Z^*_n. the entity server, who knows the corresponding private key (n, d), can easily decrypt the message, since d*c = 1 (mod Φ(n)) implies that

$$\mathbf{M} = (C^{\Phi(n)} \mod n)^{d} * \mathbf{C} \mod n \text{ ……………….equation (2)}$$

The New Proposed Scheme Performance And Security

This section analyses the mechanism, performance and security of our new proposed scheme. The new proposed scheme is a new and updated version of the original RSA.

new proposed scheme had been implemented successfully using visual basic.net. The new proposed scheme of our implementation, varying quite a few parameters that make it much complicated and therefore need more time and efforts to be break. In this research, we have considers alternative, more efficient, secure implementations of RSA with respect to industrial constraints. The new proposed scheme could be used to generate private and public key to be used as digital signature.

In RSA and surely in our new proposed scheme, decryption process is usually a lot slower than encryption since the decryption exponent is large (same size as n usually).

This somewhat changes the RSA key generation process since additional values need to be computed and stored with private key d.

The RSA Problem, as been explained above, is clearly no harder than integer factoring, since an adversary who can factor the modulus n can compute the private key (n, d) from the public key (n, e). However, it is not clear whether the converse is true, that is, whether an algorithm for integer factoring can be efficiently constructed from an algorithm for solving the RSA Problem.

Conclusion

Despite changes in the security pattern each time it is presented, our proposed scheme is sufficiently robust and efficient to be used to transfer and exchange data securely. Our scheme offers more security as it requires a genuine user's password, digital signature. Key lengths up to 1024-bit or even higher can be regenerated making the scheme compatible with the current security requirements of public key cryptosystems.

We believed that it is important to recognize that RSA and our new proposed scheme signature (encryption or decryption) performances could be significantly improved at no cost at all, provided that, much more secret.

The most widely used public key cryptosystems depend on the problem of either the factoring of the modulus, which is the product of tow large prime number , or the difficulty of calculating discrete logarithms problem . The Public-Key Encryption scheme Parameters

References;

[i] Needham, R.M., Schroeder, M.D. (1978), "**Using encryption for authentication in large networks of computers**", *CACM*, Vol. 21 No.12.

[ii] R.L. Rivest, A. Shamir, and L.M. Adleman, (1978), "**A method for obtaining digital signatures and public-key cryptosystems"**, Communications of the ACM (2) 21 , 120-126.

[iii] RSA Security (2003), "**Understanding Public Key Infrastructure (PKI)**", RSA Security Inc., available at: http://www.computel.com.lb/Downloads/PKI.pdf.

[iv] **www.pgp.com** , (2018).

[v] R.L. Rivest, A. Shamir, and L.M. Adleman, (1978), "**A method for obtaining digital signatures and public-key cryptosystems"**, Communications of the ACM (2) 21 , 120-126.

[vi] Ellison, C., Schneier, B. (2000), "**Ten risks of PKI: what you're not being told about public key infrastructure**", *Computer Security Journal*, available at: www.counterpane.com/pki-risks.pdf, Vol. XVI No.1.

[vii] RSA (2003), "***RSA Laboratories Frequently Asked Questions About Today's Cryptography",*** available at: www.rsasecurity.com/rsalabs/faq/3-1-5.html.

[viii] Stallings, W (1995), "***Network and Internetwork Security: Principle and Practice"***, Prentice-Hall, Englewood Cliffs, NJ.

[ix] A. Barker, G. Krull, and B. Mallinson, (2005) "**A Proposed Theoretical Model for M-Learning Adoption in Developing Countries**," in mLearn 2005 - 4th World conference on mLearning Cape Town, South Africa.

[x] A. Lenstra and E. Verheul, , (2000), "**Selecting cryptographic key sizes**", Proceedings of PKC 2000, Lecture Notes in Computer Science, 1751 (2000), Springer-Verlag, 446-465.

[xi] L Ecuyer, (1988), " **Efficient and Portable Combine Random Number Generator** ", ACM, V 31 n.6 , P. 742-751.

[xii] john abbot, victor shoup, Paul zimmermann ,(2000), "**factorization in Z[x]: the searching phase, ACM** , 291-276.
[xiii]J. M. DeLaurentis., (1984), "**A further weakness in the common modulus protocol for the RSA cryptoalgorithm"**. Cryptologia, 8:253–259.
[xiv] A. Lenstra and E. Verheul, (2000), "**Selecting crypto graphic key sizes", Proceedings of PKC",** Lecture Notes in Computer Science, 1751 , Springer-Verlag, 446-465.