SOCIETY FOR SCIENCE AND EDUCATION
UNITED KINGDOM

TNC TRANSACTIONS ON
NETWORKS AND COMMUNICATIONS

VOLUME 5, NO. 4
ISSN: 2054 -7420

ZRSA Algorithm

Dr. Mohamed Alzaabi
alzaabi123@gmail.com

ABSTRACT

Today, RSA algorithm is one of the encryption algorithm that is used widely. However, with the advances in computer power it is becoming susceptible to be cracked. It is become a necessity to develop a new algorithm that can withstand future processing capacity. Zaabi RSA (ZRSA) is a new algorithm that is designed to remedy RSA weaknesses.

Keywords: ZRSA; RSA; Algorithm, Cryptography; RSA; Encryption; Decryption; Key Generation

1 Introduction

Cryptography is part of the art of protecting information. The modern uses for cryptography are encrypting and decrypting emails, credit cards and other confidential data.

Cryptography is known as “symmetric key system” that generate one secret key to be shared between the sender and the receiver and “public key system” (asymmetric) that generates two keys, a public key which is known by all and a private (secret) key which is known by the receiver.

2 Encryption Techniques

While doing encryption with the help of computers, there are two main approaches. These two approaches are symmetric encryption and asymmetric encryption systems

2.1 Symmetric Encryption

Symmetric encryption, also known as secret key cryptography is based on using the same key to encrypt and decrypt a message.

This technique requires great care in key distribution because the same key both encrypts and decrypts the message [1]. The problem is how to distribute keys and the solution varies. One can get along using a key when you physically meet or remotely using any type of media. Here, it is appropriate to select a media other than the one used for the encrypted information.

2.2 Asymmetric Encryption

The form of cryptosystem that uses encryption and decryption with two separate keys is called asymmetric encryption. One of the keys is called public key and the decryption key is called private key [2].

Asymmetric encryption is thus based on a user to use different keys to encrypt and decrypt a data set. The technique is also called Public Key Cryptography.

DOI: 10.14738/tnc.54.3460
Publication Date: 23rd July 2017
URL: http://dx.doi.org/10.14738/tnc.54.3460

Each user has a key pair consisting of a public and a private key. The public key can be made available to other users via a database. The private key, as the name suggests, is available only to its owner and the user should not give it to anyone else.

Asymmetric encryption algorithms based on the use of a reverse approach are called trap door encryption [3]. A trapdoor is a process that is easy to implement in one direction, but very difficult to do if one is to go back in the other direction. The asymmetric encryption algorithms that are considered safe today use all the trap doors that are taken from mathematics.

An example of an asymmetric encryption algorithm is the RSA algorithm, to be described in section 5.1.1.2.1. It is a block cipher algorithm that divides the plaintext into blocks that are encrypted separately. The size of the blocks depends on the length of the key that is used.

It is an asymmetric encryption algorithm because it uses public and private keys. Therefore, a user must first generate public and private keys to be able to make use of the RSA algorithm. The keys are calculated according to a certain pattern to be discussed in this section.

When a message is encrypted with asymmetric technology, it is done in the following way. Subscriber A wants to send an encrypted message to subscriber B, A encrypts the message with B's public key. When B receives the message, he can decrypt it with his private key.

The RSA algorithm easily multiplies two large prime numbers p and q to power n [4] but much more difficult to factor n to p and q.

3 RSA Algorithm

RSA algorithm is an asymmetric encryption algorithm that was developed in 1977. The algorithm gets its name from its three creators' surnames, Rivest, Shamir and Adelman. It is a block cipher algorithm, divides the plaintext into blocks that are encrypted separately. The size of the blocks depends on how big the key is used.

The RSA algorithm is an asymmetric encryption algorithm that uses public and private keys. Therefore, a user must first create a public and a private key to be able to make use of the RSA algorithm. The keys are calculated according to a certain pattern.

The working Principle and the key establishment process of RSA is described as follows. The RSA Algorithm revolves around the three-basic step wise procedures and they are classified accordingly as Key Generation, Encryption and Decryption.

3.1 Key Generation

There are generally two types of keys in RSA [5]. They are classified into public and private keys. The public key can be disclosed to anyone but the private key is confidential and never disclosed as it’s primarily very important in decrypting the data encoded using the RSA system. The keys are generated from the below factors.

Initially the user opts for two distinct prime numbers p and q. For security purposes, the integers, p and q, should be chosen at random, and should be the same bit-length. Prime integers can be efficiently found using a primality test.

The next stage involves in computing and finding the value of "n" which is equal to "n=p*q"

The third iteration process is using moduli for the asymmetric cryptographic process.

The next step is computing the values of "phi" which is ϕ= (p-1) (q-1)

The Fifth step involves in choosing an integer e such that $1 < E < \phi$, $(E, \phi) = 1$

We then find the value of the $d = E^{-1}$ Mod $\phi(n)$; i.e. d is the multiplicative inverse of E mod ϕ, and d is kept as a private exponent encryption secret.

3.2 Encryption

Assume a subscriber sender at "X", transmitting their public key (n, E) to another subscriber sender at "Y", the destination user keeps the private key as secret and does not disclose any information to the system. The text message is encrypted and sent to the destination as a cipher text to the user at the receiving end:

"$C = M^E$ Mod n".

This is shown in Figure 3.2.1.

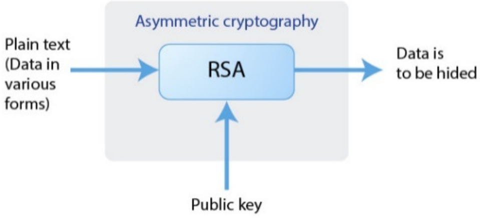

Figure 1: Encryption Using Public Key

3.3 Decryption

The Cipher text can be transformed into the original text through a recovering technique using the factor d of the cryptographic component.

The Message which is obtained can be transformed into "M = Cd Mod n”.

Generally, the length of the bit string of n should be 512 bits at least. The decryption using public at block level key is depicted Figure 3.3.1

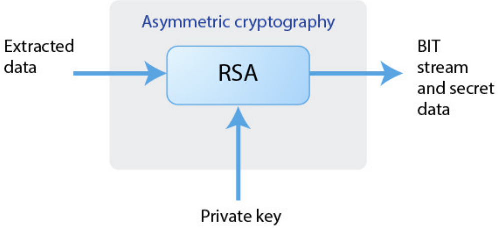

Figure 2: Decryption Using Public Key

3.4 Modified RSA with Two Random Numbers

Modified RSA with two random numbers is similar to RSA apart from using z1 and z2 to replace z.

$z_1 = x * y$ and $z_2 = x * y * a * b$. a and b are considered two random numbers. With this method, z_2 is linearly bigger than z. Hence, the modified RSA is safer than the original RSA as the de factorization would require longer time. The paper provided an example to prove its accuracy.

3.5 Modified RSA with Three Prime Numbers

Paper published by Patidar [6] listed a new RSA algorithm with three prime numbers, instead of the usual two prime numbers.

The three prime numbers are:

p, q & r.

ϕ and n are calculated as follows:

$n = p * q * r$

$\phi(n) = (p-1) * (q-1) * (r-1)$

The two conditions GCD (E, ϕ(n)) =1 and 1<E< ϕ(n) are applied to this method, which is similar to the original RSA. The sender encrypts the message using the following standard formula:

$C=M^{E}$ Mod n

While the receiver computes the private key d using:

$d = E^{-1}$ Mod ϕ(n)

The value of d would allow the receiver to decrypt the message using the following standard formula:

$M=C^{d}$ Mod n.

So, by inducing three prime numbers, this method has boosted the value of n. Since the number is bigger, this technique has met the same aim for the algorithm that uses two random numbers explained above.

3.6 Possible ways to attack RSA

The possible ways for attackers to obtain the private key d, is to view all parameters associated with the RSA algorithm and identify their weaknesses and strength.

The knowledge of the RSA algorithm is readily available to attackers/hackers. The parameters that are associated with the RSA algorithm are n, E, C and d. As disclosed above in section 3.5, n is public key and d is secret key. Worst case scenario, all other information of RSA is public, i.e. attackers may be able to get the values of n, E and C. Figure 3.6.1 exposes RSA security issues.

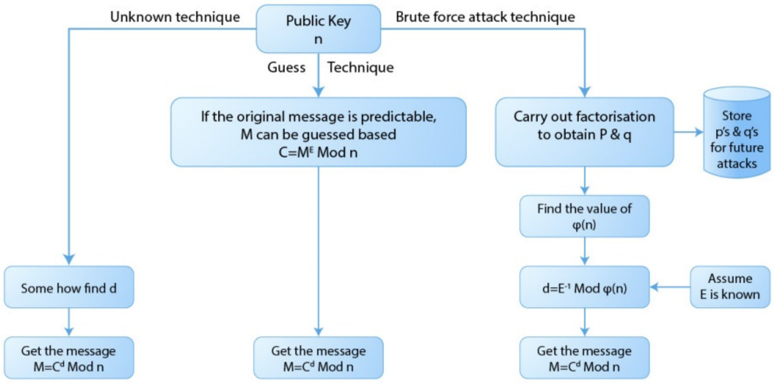
Unknown technique
Public Key n
Brute force attack technique
Guess
Technique
If the original message is predictable, M can be guessed based $C=M^{E}$ Mod n
Carry out factorisation to obtain P & q
Store p's & q's for future attacks
Find the value of φ(n)
Some how find d
$d=E^{-1}$ Mod φ(n)
Assume E is known
Get the message $M=C^{d}$ Mod n
Get the message $M=C^{d}$ Mod n
Get the message $M=C^{d}$ Mod n

Figure 3: RSA security issues

Finding d is the first technique, the Unknown Technique, which may lead to decrypting the message and publishing it in the public domain.

With the second technique, if the message is predictable, the attacker makes a guess at the original message, by giving a value to M. Therefore, M^E Mod n can be calculated. If the result matches the encrypted version, the iterative process is to be stopped, otherwise, another guess is given and the process starts again. This technique pushed computer manufacturers to append random padding to make the message unpredictable.

The third technique is Brute Force Attack. Here n is factorised to reach p & q. Hence, ϕ can be calculated. Using the Extended Euclidian algorithm, d can be calculated and the message can be decrypted. If n is large this technique could take a long time.

Figure 3.6.2 defines n as small, medium and large numbers relative to the size of bits.

n is considered as:	If:
Small	$n < 2^{136}$
Medium	$2^{136} \leq n < 2^{512}$
Large	$2^{512} < n$

Figure 4: n is defined as small, medium or large depending on the size of bits

It has been reported that, if n is small or medium, RSA has been factorised [7]. As computing power is continually increasing, it is only a matter of time for large n to be factorised. Hence, a new RSA algorithm has to be developed to make the factorisation impossible or near impossible.

4 ZRSA

The two new RSAs listed above have provided techniques to show that the value of n is increasing linearly with the introduction of new parameters. The increase in value n may make the factorisation difficult but not different from the original RSA, at least for the time being. However, time within the last two/three decades has proven that computing power is increasing continually. So, it is a matter of short time and hackers would be able to crack RSA and the above suggested RSAs, with ease.

The suggested alternative to the original RSA algorithm in this project is referred to as ZRSA. ZRSA uses three prime numbers and three random numbers which is the main difference with the original RSA algorithm. The third random number is expressed as:

$$z = e^{\text{1st Random Number}} + e^{\text{2nd Random Number}}$$

The new parameters within ZRSA affect the three components of the generated public and private keys (E, n) & (d, n), i.e. E, n & d. ZRSA Algorithm revolves around the three basic stepwise procedures and they are classified accordingly as Key Generation, Encryption and Decryption.

4.1 Key Generation

There are generally two types of keys in ZRSA, they are classified into public and private keys. The public key can be disclosed to anyone but the private key is confidential and never disclosed as it is primarily very important in decrypting the data encoded using the RSA system. The keys are generated from the below factors.

Initially, the user for ZRSA opts for three distinct prime numbers p, q and r. For security purposes, the integers, p, q and r, should be chosen at random, and should be of the same bit-length. Prime integers can be efficiently found out using a primality test. The user for ZRSA opts for two distinct random numbers a and b. The third random number labelled as z is expressed as:

$$z = e^{\text{1st Random Number}} + e^{\text{2nd Random Number}}$$

The next stage involves computing the value of "n" which is equal to "n=p*q*z"

The third iteration process is using moduli for the asymmetric cryptographic process.

The next step is computing the values of "phi" which is $\phi(n) = (p-1)(q-1)(z-1)$

The fifth step involves choosing an integer E such that $1 < E < \phi$, provided $GCD(E, \phi(n)) = 1$

The sixth step is to find the value of $d = E^{-1} \text{ Mod } \phi(n)$. From the formula d is the multiplicative inverse of E Mod ϕ. d is kept as the private exponent encryption secret.

To find the modular inverse with respect to the 'ϕ', d which is one element of the ZRSA private key, the following set of equations has to be used:

$$d = E^{-1} \text{ Mod } \phi$$

or

$$d * E = 1 \text{ Mod } \phi$$

The extended Euclidian algorithm has to be used to calculate the value of 'd', as follows:

$GCD(\phi, E) = \phi x + Ey$, where $GCD(\phi, E) = 1$ in ZRSA, and

$y = d$ if $d < \phi$ and $d \neq$ negative integer.

How to find the inverse of ϕ Mod E?

If $GCD = x*E + y*\phi$, $E = n$ and $\phi = m$, GCD in ZRSA is = 1.

Hence:

$$1 = x*E + y*\phi$$

If $E = r_{i-1}$ and $\phi = r_{i-2}$, use:

$r_{i-2} = q_i\, r_{i-1} + r_i$, q_i is quotient and r_i is the reminder ---- (1)

Equation (1) is used to generate the quotients, q_i's.

$d = x_{i-2} - (x_{i-1} * q_{i-2}) * \text{ Mod } \phi$, where $x^{-2} = 0$, $x^{-1} = 0$, $q^{-2} = 0$ & $q^{-1} = 0$

$r_{i-2} = q_i * r_{i-1} + r_i$

The above set of equations has been formalised in Excel file to produce n, E and d.

Hence, the public key and the private key are:

Public Key = (n, E) and

Private Key = (n, d)

4.2 Encryption

There are conditions to ZRSA, prior to transmitting and receiving, which have to be fulfilled to ensure security. These conditions are similar to the standard RSA conditions. The first condition is that a sender, subscriber (SS1), has to transmit a public key (n,E) to the receiver, the other subscriber (SS2). The second condition is that the destination user, SS2 keeps the private key unit d as secret and does not disclose it to the WiMAX system. The text message M is padded, encrypted and sent to the destination as a cipher text, C to the user at the receiving end. The encryption formula is:

$C = ME \text{ Mod } n$

The conceptual idea of ZRSA encryption is shown in Figure 4.2.1.

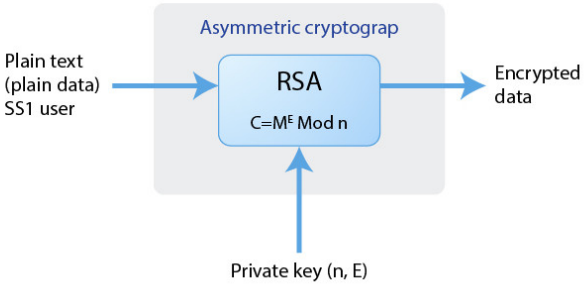

Figure 5: Encryption Using Public Key

4.3 Decryption

The Cipher text can be transformed into the original text using the factor d of the cryptographic component. The decryption formula is:

$M = C^d \text{ Mod } n$

M is transformed into plain text with a specified padding. The conceptual idea of ZRSA decryption is shown in Figure 4.3.1**.**

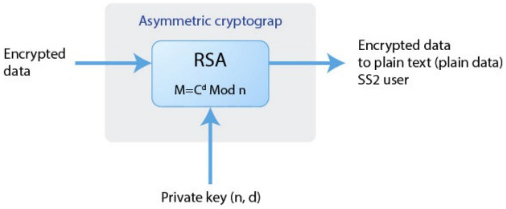

Figure 6: Decryption Using Private Key

5 Modified ZRSA

Again, the modified ZRSA is based on three basic steps: Key Generation, Encryption and Decryption. All of the three steps are described below.

5.1 Key Generation

The values of n's and ϕ's are worked out as follows:

Set of n’s equations:

$n_p = q*z$

$n_q = p*z$

$n_z = p*q$

Note that n_p is related to the combination of prime’s q & z, n_q is related to the combination of prime’s p & r and n_r is related to the combination of prime’s p & q. other combinations, such as p & q, q & z and z & p or even p & q & z, will lead to similar results.

Set of ϕ’s equations:

$\phi_p = (q-1)\ (z-1)$

$\phi_q = (p-1)\ (z-1)$

$\phi_r = (p-1)\ (q-1)$

ZRSA generates two types of keys. They are classified into public and private keys. The public key can be disclosed to anyone but the private key is confidential and never disclosed as it is primarily very important in decrypting the data encoded using the RSA system. The keys are generated from the below factors.

Initially, the user for ZRSA opts for three distinct prime numbers p, q and z. For higher security purposes, the integers, p, q and z, should be chosen at random and the same bit-length. Prime integers can be efficiently found out using a primality test. The user for ZRSA opts for two distinct random numbers, namely a & b. The third random number labelled as z expressed as the multiplication of the first two random numbers, a & b. As the random numbers will increase the value of n, within this section, they have been eliminated from the following example.

The next stage involves computing the value of "n_p, n_q & n_z “.

The third iteration process is using moduli for the asymmetric cryptographic process.

The next step is computing the values of "phi’s" which are” ϕ_p, ϕ_q & ϕ_z “.

The fifth step involves choosing an integer E such that $1 < E < \phi$, GCD (E, ϕ(n)) = 1

The sixth step is to find the value of $d = E^{-1}$ Mod ϕ(n). From the formula, d is the multiplicative inverse of E Mod ϕ. d is kept as the private exponent encryption secret.

To find the modular inverse with respect to 'ϕ', d, the following set of equations has to be used:

$d_p * E_p = 1 \text{ Mod } \phi_p$

$d_q * E_q = 1 \text{ Mod } \phi_q$

$d_z * E_z = 1 \text{ Mod } \phi_z$

The extended Euclidian algorithm has to be used to calculate the value of 'd', as follows:

GCD (ϕ, E) = ϕx + Ey, where GCD (ϕ, E) = 1 in ZRSA, and

$y = d$ if $d < \phi$ and $d \neq$ negative integer.

How to find the inverse of ϕ mod E for p, q & z primes?

If GCD = $E*x + \phi*y$, E = n and ϕ = m, GCD is equal 1.

Hence:

$1 = E*x + \phi*y$

If $E = r_{i-1}$ and $\phi = r_{i-2}$, use the following iterative equations:

$r_{i-2} = q_i\, r_{i-1} + r_i$ q_i is quotient and r_i is the remainder

$d = x_{i-2} - (x_{i-1} * q_{i-2}) * \text{Mod } \phi$, and

$r_{i-2} = q_i\, r_{i-1} + r_i$

The above set of equations has produced n, E and d. Hence, the public key and the private keys for primes p, q and z are:

p

Public Key = (n_p, E_p) and

Private Key = (n_p, d_p)

q

Public Key = (n_q, E_q) and

Private Key = (n_q, d_q)

r

Public Key = (n_z, E_z) and

Private Key = (n_z, d_z)

ZRSA public and private keys can be summarised as two set of keys:

Public key (E_p, E_q, E_z, n_p, n_q, n_z)

Private key (d_p, d_q, d_z, n_p, n_q, n_z)

p, q & z keys are used for successive letters repeatedly.

5.2 Encryption and Decryption

The encryption and decryption keys for:

p

To be applied to the first letter of the encrypted/decrypted message:

Encryption $C = M^{Ep} \pmod{n_p}$

Decryption $M = C^{dp} \pmod{n_p}$

q

To be applied to the second letter of the encrypted/decrypted message:

Encryption $C = M^{Eq} \pmod{n_q}$

Decryption $M = C^{dq} \pmod{n_q}$

z

To be applied to the third letter of the encrypted/decrypted message:

Encryption $C = M^{Ez} \pmod{n_z}$

Decryption $M = C^{dz} \pmod{n_z}$

6 Conclusion

With the continuous advances in computing power, RSA algorithm is becoming subject to be cracked soon. However, the new cryptography algorithm ZRSA will ensure that cracking is not achievable in the near future since it is almost impossible to break ZRSA algorithm due to the way its secret keys are calculated.

REFERENCES

[1]. Delfs, H. &. K. H., 2007. *Introduction to cryptography: principles and applications*. Canada: Springer. .

[2]. Ferguson, N. & Schneier, B., 2003. *Introduction to cryptography: principles and applications*. NY: Wiley.

[3]. Diffie, W. & Hellman, M., 1976. *New Directions in Cryptography*. *IEEE TRANSACTIONS ON INFORMATION THEORY,* VOL. IT-22(6), pp. 644-654.

[4]. Li, Y., Z, Y. & Nui, W., 2010. *A Method of Privacy Preserving in Mobile Wireless Environments.* s.l., 7th International Conference on In Ubiquitous Intelligence & Computing and 7th International Conference on Autonomic & Trusted Computing (UIC/ATC), IEEE.

[5]. Ferguson, N. & Schneier, B., 2003. *Introduction to cryptography: principles and applications*. NY: Wiley.

[6]. Patidar, R. a. B. R., 2013. *Modified RSA Cryptosystem Based on Offline Storage and Prime Number*. *2013 IEEE International Conference on Computational Intelligence and Computing Research.*

[7]. Al-Hamami, A. H. &. A. I. A., 2012. *Enhanced Method for RSA Cryptosystem Algorithm*. *2012 International Conference on Advanced Computer Science Applications and Technologies,* pp. 402-408.