Public Key Encryption: The RSA Algorithm

Lynne Cheravitch, Jillian Gaietto, Jessica Pugliese
Advisor: Dr. Gang Yu, Professor of Mathematics, Kent State University

Abstract

Over the past decade, the frequency and sophistication of intrusions into U.S. government, private industries and personal databases has grown exponentially. As the scale of cyber security threats tips to threaten national security, the need for layered and sturdy defenses to protect vital networks and infrastructure is growing. One of the most successful public key encryption methods is the RSA Algorithm, which utilizes the mathematical difficulty of factoring the product of two prime numbers. Our goal is to provide interested parties with an understanding of how the RSA Public Key Algorithm works, and how it benefits and protects our information technology-dependent society. Our combined knowledge was extracted from declassified written materials, consultation with our supervising professor Dr. Gang Yu, and experience working with government agencies.

Setting Up the Algorithm

Step 1. We need to have 2 large distinct prime numbers. We call these p and q.

Step 2. We find n=p*q

Step 3. We need to find

phi(n)=φ(n)=(p-1)*(q-1)

Step 4. We need to choose an integer e, 1<e< φ(n) such that gcd(e, φ(n))=1 (i.e. e and φ(n) are relatively prime).

Step 5. Finally we need to generate the "secret" number d, 1<d< φ(n), satisfying d*e≡1$^{(mod\ \varphi(n))}$

Then we have: (n,e) as the public key and (d,p,q, φ(n)) as the private information.

- When we put RSA Encryption into action, we will also denote the message that needs encoded as:

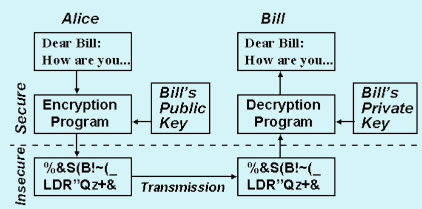
C≡M$^{e\ (mod\ \varphi(n))}$ where C is the encrypted number and M is the message we are trying to recover.

History of the Algorithm

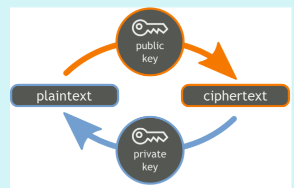
The concept of public-key encryption was discovered in 1976 by Whitfield Diffie and Martin Hellman. Diffie and Hellman created a method to send messages securely with a public key to the cryptographic system. They were unable however, to create the method mathematically. One year later, three professors from MIT developed the algorithm mathematically and essentially created one of the most sophisticated public key encryption methods still in use today. Those three professors were Ron Rivest, Adi Shamir and Leonard Adlemen, hence the name, RSA Algorithm.

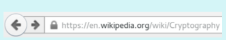
Rivest, Shamir and Adlemen used the concept of Diffie and Hellman and produced an algorithm using a one-way function. That is, a function that is easy to compute one way but is nearly impossible to compute in reverse. The basis for the algorithm begins with two large prime numbers. It is easy to compute the product of these two prime numbers but it is nearly impossible to factor the product of these two numbers without knowing the original primes.

In public key encryption, there is no need to exchange a key between two parties before a message is sent. Party B creates an encryption key and a decryption key. B keeps the decryption key private and he publishes the encryption key for all to see. Then, when party A wants to send a message to party B, A uses the encryption key published by B. Then party B decrypts the message using the confidential encryption key that no one knows but him. Here, the encryption key is the product of our two large prime numbers and our decryption key begins with the two large prime numbers. Thus, calculating the decryption key is extremely difficult and could take years to compute. Therefore, using the RSA Algorithm, no key exchange needs to take place between the two parties and transmitting the message is extremely secure.



RSA Encryption's Creators





Above: Padlock icon from the Firefox Web browser, which indicates that TLS, a public-key cryptography system, is in use.

Left: RSA encryption is used today by constantly switching security keys on devices

The Algorithm

Let our parties A and B, be referred to as Alice and Bob.

A. First Bob creates the encryption and decryption keys as follows:
1) Bob chooses secret prime numbers p and q and computes their product n=p*q. Typically p and q are several hundred digits in length.
2) Bob chooses an integer e, such that gcd(e, (p-1)(q-1)) = 1. (Note- (p-1)(q-1) = φ(n))
3) Bob computes d such that de≡1 mod φ(n)
4) Bob publishes (n,e) and he keeps (d,p,q, φ(n)) private.

A. When Alice is ready to send Bob a message, she does the following:
1) Alice takes the message and converts it into a number M.
2) She takes n and e that Bob published and creates a *cyphertext* by computing C≡M^e (mod n)
3) Alice sends C to Bob.

A. Bob decrypts the message C by doing the following:
1) Bob computes M≡C^d (mod n)
   a. Note - C^d (mod n) is congruent to the number M based on a theorem from number theory known as Euler's generalization of Fermat's Little Theorem.
2) Bob converts the number M back into the message.

An outside party would have the cyphertext C and the encryption key n and e. However, this third party would need the number d to decrypt the message. Since d is computed using e, p, and q, the third party would need the numbers p and q to decode the message. In order to get p and q, you need to factor the number n. And remember p and q are hundreds of digits in length. Thus, factoring n into p and q is nearly impossible and could take years to compute using the fastest computers on Earth. This is why this algorithm is extremely useful and very powerful.

RSA Encryption in Action

(example provided by Dr. Gang Yu, Professor of Mathematics, Kent State University).

We need to let the letters of the alphabet be denoted by numbers, i.et, 00 – "Blank"; 01 – "a"; 02 – "b" … 25 – "y"; 26 – "z"

Recently, Alice and Bob have been sending message to each other using the RSA Algorithm. Their public key is n=338,699 and e=77,893, and only Bob knows that n=p*q and p=577, q=587, thus n=577*587. Alice was accepted to graduate school and Bob asked what school Alice would be attending. Alice answers C=223,208. What is the graduate school Alice will be attending?

1) First we need to find φ(n)= φ(p*q)=(577-1)(587-1) = 337,536
2) Using the Euclidean Algorithm we will find our number d satisfying d*e≡1$^{(mod\ \varphi(n))}$.

φ(n)=e*b + r

(Where b is an integer and r is the remainder)

337,536 = 77,893*4 + 25964

77,893 = 25,964*3 + 1
1 = 77,893 – 3(25,964)
= 77,893 – 3*[337,536-(4)77,893]
= e – 3*[φ(n)-4*(e)]
= 13e – 3*φ(n)

Hence, 1 = 13e – 3*φ(n) and equivalently, 13e≡1$^{(mod\ \varphi(n))}$.

So d = 13.

3) Our message C=223,208 can be written C≡M$^{e\ (mod\ n)}$. Thus,

C^t≡ (M^e)$^{t\ (mod\ n)}$ ≡ M$^{(3*\varphi(n)+1)\ (mod\ n)}$ ≡ M$^{(\varphi(n))*3}$ M$^{(mod\ n)}$ ≡ (1)^{3}M$^{(mod\ n)}$ ≡ M$^{(mod\ n)}$

Therefore, M ≡ C$^{t\ (mod\ n)}$

C^t≡ (223,208)^{13} (mod 338,699)

(Note: 13 = 8 + 4 + 1 = $2^3 + 2^2$ + 1)

M ≡ C^{13} ≡ C^8 * C^4 * C^1

- Find C^1, C^2, C^4, C^8

C^1 = (223,208)^1 ≡ 223208 $^{(mod\ n)}$

C^2 = (223,208)^2 ≡ 204,461 $^{(mod\ n)}$

C^4 = (223,208)^4 ≡ 37747 $^{(mod\ n)}$

C^8 = (223,208)^8 ≡ 268015 $^{(mod\ n)}$

- M ≡ (268,015)*(37,747)*(223,208) $^{(mod\ n)}$

M ≡ (161,774)*(223208) $^{(mod\ n)}$

M ≡ 211902.9988 ≈211,903

Evaluating these numbers using our alphabet key, we get 03=C, 19=S, 21=U. We encrypt the message backwards, as an integer in our calculations would not begin with a zero. That is, if our answer were 1231402, it is translated 02=b, 14=n, 23=w and ***1=01=a***. Our answer would not yield 01231402 as we are dealing with integers.

Therefore, Alice will be attending USC for graduate school!

Applied Proof

To define the RSA Algorithm as a general cryptographic algorithm, there must be a general proof for any message *m* that must be encrypted. **Suppose:**

GCD(*p*, *q*)= 1
N=*pq*
ed=1 *mod ϕ*(*N*)

Claim: (*me*)*d*=*m mod(N),* ∀ *m* ∈ *Zn*

Proof:
Being *m* ∈ *Zn* there are only two possible cases to analyze:

1) *GCD*(*m*, *N*)= 1

In this case Euler's Theorem stands true, assessing that *mϕ* (*N*)=1(*mod N*). As for the claim to prove, because of the third condition, we can write: (*me*)*d*=*med*=*m*1+*kϕ*(*N*), Furthermore, *m*1+*kϕ*(*N*)=*m*∗*mkϕ*(*N*)=*m*∗(*mϕ*(*N*))*k*, and for Euler's Theorem *m*∗(*mϕ*(*N*))*k*=*m*(*mod N*). Proving that the thesis stands in this case.

2) *GCD*(*m*, *N*)≠ 1

In this case Euler's Theorem does not stand true any more. By the Chinese Remainder Theorem, it is true that if *GCD*(*p*, *q*)=1 then: *x*=*y*(*mod p*)∧*x*=*y*(*mod q*)⇒*x*=*y*(*mod pq*) So by proving the following two statements we would have finished: (*me*)*d*=*m modp*, (*me*)*d*=*m modq*. Since *GCD*(*m*, *N*)≠1 between *GCD*(*m*, *N*)=*p*, and *GCD*(*m*, *N*)=*q* must stand true. Next we must demonstrate that both the above statements stand true in the case *GCD*(*m*, *N*)=*p*, being it absolutely identical (by switching letters) to prove it for *GCD*(*m*,*N*)=*q* as well. So let it be *GCD*(*m*, *N*)=*p*, this implies that *m*=*kp* for some *k*>0 which means that *m(modp)*=0. By concerning the first statement we also have (*me*)*d*=((*kp*)*e*)*d* which therefore results to be a multiple of *p*, and so it is equal to zero. So the first statement becomes 0=0 and is proven to be satisfied. Concerning the second statement we have that Euler's Theorem results to be proved in *Zq* since *GCD*(*m*,*q*)=1, so: *mϕ*(*q*)=1(*modq*). This implies that we can write: (*me*)*d*=*med*=*med*−1*m*=*mh*(*p*−1)(*q*−1)*m*=(*mq*−1)*h*(*p*−1)*m*=1*h*(*p*−1)*m*=*mmodq*. which definitively proves the second statement and theorem.

Friendly Reminder:

How to find (223208)^2 ≡ 204,461 $^{(mod\ n)}$
1) Take (223208)^2 / 338699 = 147097.6037
2) Subtract the integer and multiply the decimal by n

147097.6037-147097=.6037

.6037*338699 = 204460.9994 ≈ 204461

Resources: The RSA Algorithm. (n.d.). Retrieved March 24, 2015, from http://facultyfp.salisbury.edu/despickler/personal/Resources/TechnologyWorkshops/ScienceNight2013/RSAHandout.pdf Photo taken from http://imps.mcmaster.ca/courses/SE-4C03-07/wiki/wrighd/rsa_alg.html The RSA Algorithm. (n.d.). Retrieved March 24, 2015, from http://facultyfp.salisbury.edu/despickler/personal/Resources/TechnologyWorkshops/ScienceNight2013/RSAHandout.pdf