The RSA Encryption Algorithm

John Paul S. Guzman

Mathematics Department, De La Salle University, 2401 Taft Ave., Manila 0922, Philippines

Abstract

Encryption is the process of scrambling information in such a way that it can only be read by select individuals. It is an essential tool as we rely on the internet to accomplish various activities such as online banking, transactions, and private messaging. In these activities, we are often required to communicate sensitive information like passwords, credit card information, home addresses, etc. This paper discusses the RSA encryption algorithm, its mathematical foundations, and how it is used to protect sensitive information.

Keywords: Cryptography, Internet security, Prime numbers.

1 Introduction

The RSA encryption algorithm is a widely used encryption algorithm that secures sensitive data from "man-in-the-middle" attacks wherein malicious third-party eavesdrops on the communication between two parties with the intent of stealing sensitive information [1]. It is one of the earliest implementations of asymmetric encryption which allows users to establish a private communication over a public channel [2]. This is accomplished by having separate keys for encryption (public key) and decryption (private key). On the other hand, symmetric encryption uses the same key for encryption and decryption.

Another key feature of RSA is the ability to create digital signatures. Similar to a physical signature, an digital signature provides a way to prove that a message originated from the sender along with the means to show that the message was not tampered with [2]. RSA uses Euler's theorem to: (a) scramble or encrypt a message into ciphertext, (b) unscramble or decrypt ciphertext back to the plaintext, (c) generate digital signatures.

2 Cryptographic Operations

In order to create the public key and private key, we need to choose two large distinct prime numbers p and q at random. Let n be the product of p and q. Next, pick a large random integer d that is relatively prime to $\phi(n)$ where ϕ is the Euler totient function. Finally, we choose e to be the multiplicative inverse of d modulo $\phi(n)$.

$$ed \equiv 1 \mod \phi(n).$$

We know that such an e is exists since d is relatively prime to $\phi(n)$. The pair (e, n) will serve as the public key, while the pair (d, n) will serve as the private key.

Let M be a plaintext message encoded as a number. We insist that $M < n$. In the case where $M \geq n$, we will split M into multiple chunks and send them separately. Furthermore, we insist that M is relatively prime to n. In practice, this is a safe assumption since it is unlikely that M is a multiple of p or q. Regardless, we could easily check for this case since we have access to p and q. If so, we can simply split M further or add some padding until it becomes relatively prime to n.

We generate the ciphertext C by raising M to eth power then taking modulo n. We can decipher C to the plaintext D by raising C to the dth power then taking modulo n. Lastly, we can create the digital signature S by raising M to the dth power then taking modulo n.

$$C \equiv M^e \mod n.$$
$$D \equiv C^d \mod n.$$
$$S \equiv M^d \mod n.$$

3 Mathematical Foundations

To illustrate the correctness of these of operations, we first need to discuss some results in number theory.

Theorem 3.1 *If $a \equiv b \mod n$, then $ca \equiv cb \mod n$ and $a^k \equiv b^k \mod n$ for any positive integer k.*

Theorem 3.2 *If a and b are relatively prime with $a \mid c$ and $b \mid c$, then $ab \mid c$.*

Corollary 3.2.1 *If m and n are relatively prime with $a \equiv b \mod m$ and $a \equiv b \mod n$, then $a \equiv b \mod mn$.*

Theorem 3.3 *If p is prime, then $\phi(p) = p - 1$.*

Theorem 3.4 *If a and b are relatively prime, then $\phi(ab) = \phi(a)\phi(b)$.*

Theorem 3.5 (Euler's theorem [3]) *If a is relatively prime to b, then $a^{\phi(b)} \equiv 1 \mod b$.*

We can now show that decrypting the ciphertext does indeed yield the original message. By the choice of e and d,

$$ed = k\phi(n) + 1 \text{ for some integer } k. \tag{1}$$

Since p and q are distinct primes, we can use 3.3 and 3.4 to obtain

$$\phi(n) = \phi(pq) = \phi(p)\phi(q) = (p-1)(q-1). \tag{2}$$

Since M and p are relatively prime, we can apply 3.5 to obtain

$$M^{\phi(p)} \equiv M^{p-1} \equiv 1 \mod p. \quad (3)$$

Furthermore, by 3.1,

$$M^{k\phi(n)} \equiv (M^{p-1})^{k(q-1)} \equiv 1^{k(q-1)} \equiv 1 \mod p. \quad (4)$$

$$M^{k\phi(n)+1} \equiv M(1) \equiv M \mod p. \quad (5)$$

We can apply similar arguments for q,

$$M^{k\phi(n)+1} \equiv M \mod q. \quad (6)$$

Since p and q are distinct primes, we can apply 3.2.1 to 5 and 6.

$$M^{k\phi(n)+1} \equiv M \mod n. \quad (7)$$

The desired result is obtained by 1 and 7.

$$D \equiv C^d \equiv (M^e)^d \equiv M^{ed} \equiv M^{k\phi(n)+1} \equiv M \mod n. \quad (8)$$

A similar result is used in the implementation of signatures. This states that decrypting the message then encrypting it will also result in the original message.

$$S^e \equiv (M^d)^e \equiv M^{ed} \equiv M \mod n. \quad (9)$$

4 Application in RSA

We illustrate the application of these results in the following scenario. Suppose that Alice wants to send a message M to Bob. Bob will generate his public key (e_B, n_B) and private key (d_B, n_B), then publish the only public key. Alice would take Bob's public key, use it to create the ciphertext $C \equiv M^{e_B} \mod n_B$, then send it over. Bob can then recover the original message by $M \equiv C^{d_B} \mod n_B$ as in 8. Now suppose Eve was eavesdropping in their conversation. This means that she would have a copy of C and the public key (e_B, n_B). However, she will be unable to decipher C since she does not have access to d_B. It has been conjectured in [1] that finding d_B from (e_B, n_B) requires an exhaustive search, and so it is unfeasible for a sufficiently large value of n_B.

Hence, we are successful in establishing a private conversation over a public channel. This bypasses one of the biggest problems in symmetric encryption, namely, key distribution. This issue is due to the fact that two parties have to agree on which keys to use for encryption and decryption. If Alice and Bob were to use symmetric encryption, then Eve could simply make a copy of the key during the phase where Alice and Bob were exchanging keys. Eve can then decipher all future communication since the same key is used for both encryption and decryption.

Next, we consider the scenario where Alice wants to send a signed message M to Bob. Alice can create a signature for this message by $S \equiv M^{d_A} \mod n$. She can send S as an encrypted message to Bob as in the previous scenario, i.e., S is encrypted by Alice with Bob's public key, then decrypted by Bob with his private key. Bob can recover the message from the signature by $M \equiv S^{e_A}$ as in 9.

We can confidently say that the message came from Alice since only she has access to d_A, and so only she could have generated the signature S. Moreover, we can be confident that Bob did not modify the message M into a tampered message M'. This is due to the fact that he is unable to produce the corresponding signature S' for M' since he does not have access to d_A. In other words, his knowledge of both M and S cannot be used to created forgeries.

5 Conclusions

to do

References

[1] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, vol. 21, no. 2, pp. 120–126, 1978.

[2] W. Diffie and M. Hellman, "New directions in cryptography," *IEEE transactions on Information Theory*, vol. 22, no. 6, pp. 644–654, 1976.

[3] D. M. Burton, *Elementary number theory*. Tata McGraw-Hill Education, 2006.