RSA Cryptosystem

Computational Essay

Karim El Shenawy

March 2021

Contents

1 Introduction

In the era of modern cryptography, more often then ever, public key cryptography or asymmetric cryptography rose to popularity due it's superiority in-terms of security over symmetric cryptography. The advantage is due to less risk of man-in-the-middle attack in complex public key encryption scenarios. Public key cryptography consists of an encryption algorithm where users hold a pair of keys, a public key (which is made known to everyone) and a private key (which is remain a secret only to its respective owner). Using Public and Private keys, an encryption key is created and made public while the decryption key is also created but is made private. This method of the generating secure keys pairs is done through a mathematical problem that allows this property.

At the Massachusetts Institute of Technology (MIT) in 1977, one of the oldest and widely used public-key cryptosystem was first publish, **RSA**, named after it's designers, Ron **R**ivest, Adi **S**hamir, and Leonard **A**dleman. The RSA uses the mathematical *factorial* problem in order to preserve it's security. The encryption relies on the difficulty of factorization of two large prime numbers. In addition, RSA uses the Modular Multiplicative Inverse property to generate the pair of keys. The patent for RSA was granted on September 20^{th} 1983 to MIT and it was set to expire on September 21^{st} 2000 but was voluntarily released on September 6^{th} of that same year. This cryptosystem is can be found used in digital signature authentication and thus in digital certificates. In fact, it was used with Transport Layer Security (TLS) protocol, Virtual Private Networks (VPNs), email services, web browsers and in others products and algorithms like *the Pretty Good Privacy* algorithm. However, since the RSA is relatively slow compared to other ciphers, it is mostly used in key distribution of symmetric keys. RSA is still used to this day, however it has been slightly replaced or built on for improvement as several vulnerabilities have been found.

2 RSA Overview

2.1 Methodology

As previously mentioned, the RSA encryption is a public key cryptosystem with a pair of keys used for encryption and decryption. As well as, it uses the factorial problem to maintain it's security. The RSA encryption works as follows:

1. Consider two distinct primes p and q thus $p \neq q$ where both primes are a secret.
2. Calculate n $\implies n = p \times q$. The value of n will be made public.

3. Calculate the Euler Totient Function of n $\implies \phi(n) = (p-1) \times (q-1)$.

4. Compute an integer E such that $gcd(E, \phi(n)) = 1$ where $1 < E < \phi(n)$. E will be set to public, thus making the **Public key =** $\{n, E\}$.

5. Calculate integer D where $1 < D < \phi(n)$ such that $D \equiv E^{-1} \pmod{\phi(n)}$ or $ED \equiv 1 \pmod{\phi(n)}$. D will be set to private, thus making the **Private key =** $\{n, D\}$.

6. To Encode, we consider a plaintext P, $0 < P < n$, and we calculate the ciphertext C with **the encoding exponent E**. Therefore, $C \equiv P^E \pmod{n}$.

7. To Decode, we consider a ciphertext C and we calculate the plaintext P with **the decoding exponent D**. Therefore, $P \equiv C^D \pmod{n}$.

This procedure holds behinds the RSA principle that for some natural number W,

$$(W^E)^D \equiv (W^E)^D \equiv W \pmod{n}.$$

The proof of the correctness of RSA in it's first publication was done using *Fermat's little Theorem* which states that *If p is a prime and a is an integer relatively prime to p, then* $a^{(p-1)} \equiv 1 \pmod{p}$. We will also need to establish a few more Theorems (found in the Annex in Section 4)

Proof. Given positive integer n and primes p and where $n = p \times q, p \neq q$. Suppose there exist number E such that $gcd(E, \phi(n)) = gcd(E, (p-1)(q-1) = 1$. Suppose there also exists a number D such that $ED \equiv 1 \pmod{\phi(n)}$. Say for some natural number W, $(W^E)^D \equiv (W^E)^D \equiv W \pmod{n}$. Thus by direct proof,

$$\begin{aligned} W^{ED} &\equiv W \pmod{pq} \\ W^{1+y(p-1)(q-1)} &\equiv W \pmod{pq} && \textbf{By Theorem 3 in section 4.3} \\ W &\equiv W \pmod{pq} && \textbf{By Theorem 2 in section 4.2} \end{aligned}$$

□

Similarly, we can also prove RSA using Euler's Theorem which states that *If a and n are integers with* $n > 0$ *and* $(a, n) = 1$, *then*

$$a^{\phi(n)} \equiv 1 \pmod{n}$$

Thus proof of the correctness of RSA can be as follows;

Proof. Given positive integer n and primes p and where $n = p \times q, p \neq q$. Suppose there exist number E such that $gcd(E, \phi(n)) = gcd(E, (p-1)(q-1) = 1$. Suppose there also exists a number D such that $ED \equiv 1 \pmod{\phi(n)}$. This also suggests that $ED = 1 + x\phi(n), \exists x \in \mathbf{Z}$. Now, say for some natural number W, $(W^E)^D \equiv (W^E)^D \equiv W \pmod{n}$.Thus by direct proof,

$$\begin{aligned} W^{ED} &\equiv W \pmod{pq} && \\ W^{1+x\phi(n)} &\equiv W \pmod{pq} && \textbf{Since } ED = 1 + x\phi(n) \\ W^1(W^{\phi(n)})^x &\equiv W \pmod{pq} && \\ W^1(1)^x &\equiv W \pmod{pq} && \textbf{By Euler's Theorem} \\ W &\equiv W \pmod{pq} && \end{aligned}$$

□

2.2 Encryption/Decryption of Messages

With the publication of RSA, sending encrypted messages became more secure and feasible. However, this entails a method of converting letters and sentences into numerical values to be accepted in the RSA cryptosystem. there already exists several methods of converting letters into numbers. One of which is to represent each letter in the alphabet with a number. For instance, A can be 1, B can be 2 and so on until Z which is 26. Although this method is feasible, it does not involve the sort of numerical representation required in an RSA cryptosystem. Hence, we would need to find more sophisticated methods.

Below are a few possible methods that can be accepted by the RSA cryptosyste:

- ASCII character values by either taking the characters decimal or hexadecimal values. For example, A is 65 in decimal and 41 in hexadecimal. The advantage of this is that each lower and capital case letters have their own values.
- The form of a custom reference table with 3 columns each holding a maximum of 10 letters. Thus when we reference a letter, we will need to record it's column and row in the table. However, this method takes a long time with encryption as we also must encode the row and column to each character.
- Using other ciphers to encrypt the message into numbers is also feasible, like the Nihilist cipher. This type of cipher falls under symmetric encryption thus we would require a key. The Nihilist cipher

constructs a Polybius square using mixed alphabet using our key word to establish letter positioning in the square. Now to encrypt, we can map out each letter with it's column and row value which will result into a 2 digit number for each character in our plaintext.

It is important to know which method is used before encoding or else decryption won't be successful. Moreover, using the first method from the list, a simple example of the RSA encryption can be as follows:

1. p = 31 and q = 67
2. $n = p \times q = 2077$
3. $\phi(n) = (p-1) \times (q-1) = 31 \times 66 = 1980$.
4. Since we need E such that $gcd(E, \phi(2077)) = 1$ where $1 < E < \phi(2077)$. There are several values that hold. For simplicity let's choose 7. Thus making the **Public key =** $\{2077, 7\}$.
5. We need D where $1 < D < \phi(2077)$ such that $D \equiv 7^{-1} \pmod{\phi(2077)}$ or $7D \equiv 1 \pmod{\phi(2077)}$. D can then possibly be 283. Thus making the **Private key =** $\{2077, 283\}$.
6. Suppose we want to encode the plaintext "Maths". In ASCII, each letter maps to a value, we can then encode each letter.
   - "Cat" = 67, 97, 116
   - For abstraction and simplicity, let the value above be **P**.
7. To Encode, plaintext P and we calculate the ciphertext C with **the encoding exponent 7**.
   - "C": $67^7 \equiv 463 \pmod{1980}$
   - "a": $97^7 \equiv 1753 \pmod{1980}$
   - "t": $116^7 \equiv 1196 \pmod{1980}$
   - Thus our encoded message is 5,13,2.
8. To Decode, we consider the encoded message and we calculate the plaintext P with **the decoding exponent 283**.
   - $463^{283} \equiv 67 \pmod{1980}$ mapping it with ASCII will result to "C"
   - $1753^{283} \equiv 97 \pmod{1980}$ mapping it with ASCII will result to "a"
   - $1196^{283} \equiv 116 \pmod{1980}$ mapping it with ASCII will result to "t"

.

2.3 Security & Vulnerability

Over the course of years, RSA has proven to be useful in several products and algorithms. However, several vulnerabilities have been shown in specific attacks against RSA which are mathematical but also computational.

2.3.1 Prime Factorization & Weak Random Primes

To begin with, an obvious mathematical attack is *Prime Factorization*. Since in the RSA crypotsystem, n is public and it is composite of the 2 distinct private primes p and q. Thus a successful factorization of n can expose p and q hence consequentially exposing the entire encryption. However, this method is very exhaustive and requires a lot of computational power when it comes to a large n (large p and q as well). For this reason, in practice, p and q are usually extremely large prime values. By Infinitude pf Primes Theorem, in section 4.5, we know that there a infinitely number of primes. Now the issue is find large primes, this leads to theorems such as the Proth's Test, Pepin's Test, Fermat's little Theorem and Lucas-Lehmer Test which test the primality of numbers.

With Quantum Computers being developed, prime factorization is the biggest threat to RSA encryption. In fact, there already exists an algorithm that can successfully perform prime factorization in milliseconds. This algorithm is called *Shor's algorithm*. It is in fact a quantum computer algorithm for integer factorization that runs in polynomial time. Therefore, prime factorization is the most threatening attack against the RSA cryptopsystem.

There also exists another risk to RSA, this time it is not related to n but rather p and q. Specifically, the proximity of these 2 values. In fact, if $p - q < 2n^{0.25}$, then solving for p and q from n becomes trivial as if $p - 1$ or $q - 1$ has small prime factors, n can be easily factored using Pollard's p-1 algorithm. Thus, again the choice of p and q are extremely important in determining the security of the RSA cryptosystem.

2.3.2 Timing & Side Channel Attacks

Another successful attack against RSA is Timing attacks. This type of attack is computational which requires the analysis of the computer hardware that performs decryption. Since, for RSA to be secure, it requires large prime values. This entails large calculations that need to be compute with a computer. In 1995, Paul C. Kocher discovered this attack and in 2003, it was proven to be successful over network analysis. This attack consists of analysing the decryption time a computer takes over the course of multiple

decryption operations. This helps deduce the decryption key through probability. However, all ciphertexts must be known for this to succeed. To defend against this attack, computers must ensure that decryption take a constant time as well as implementing cryptographic binding to RSA where the encryption method takes a random number r and the decryption method would result to $(r^E C)^d \equiv P \pmod{\phi(n)}$. By Euler's Theorem, we can conclude that it is the equivalent of $(rC)^d \equiv P \pmod{\phi(n)}$.

There also exists several other side channel attacks such as the analysis of the processors used for decryption by analysing their branch predictor. However, this is more of a computational flaw of the system where the RSA decryption is performed on.

3 Conclusion

Altogether, the RSA cryptosystem has held its position as the oldest and most popular public key encryption since it's publication in 1977. Capitalizing on the factorial problem, it creates encoding exponent and a decoding exponent that are kept, respectively in the public and private key. Contributing to several products and algorithms, RSA is known for it's usage in key distribution with AES encryption. Although, the algorithm is threatened by both mathematical and computational attacks, RSA is regarded as a highly dependant on the prime values it requires. Nevertheless, RSA is well known and currently used all over the world.

4 Annex

4.1 Theorem 1

If p and q are distinct prime numbers and W is a natural number with $(W, pq) = 1$, then $W^{(p-1)(q-1)} \equiv 1 \pmod{pq}$.

Proof. Given p and q are distinct prime numbers and W is a natural number with $(W, pq) = 1$, we can deduce that $(W, p) = (W, q) = 1$, by Theorem 4.29. With that we can apply the Fermat's Little Theorem to showcase that $W^{p-1} \equiv 1 \pmod{p}$ and $W^{q-1} \equiv 1 \pmod{q}$. This also implies that $W^{(p-1)(q-1)} \equiv 1 \pmod{p}$ and similarly for $W^{(p-1)(q-1)} \equiv 1 \pmod{q}$. Therefore, by the Chinese Remainder Theorem, we can conclude that $W^{(p-1)(q-1)} \equiv 1 \pmod{pq}$. □

4.2 Theorem 2

If p and q are distinct prime numbers, k be a natural number, and W be a natural number less than pq. Then,

$$W^{1+k(p-1)(q-1)} \equiv W \pmod{pq}.$$

Proof. By direct proof, given p and q are distinct prime numbers, k be a natural number, and W be a natural number less than pq which implies that $(W, pq) = 1$;

$$\begin{aligned}
W^{1+k(p-1)(q-1)} &\equiv W \pmod{pq} \\
W^1 W^{k(p-1)(q-1)} &\equiv W \pmod{pq} \\
W^1 (W^{(p-1)(q-1)})^k &\equiv W \pmod{pq} \\
W^1 \cdot 1^k &\equiv W \pmod{pq} && \textbf{By Theorem 1} \\
W^1 &\equiv W \pmod{pq} \\
W &\equiv W \pmod{pq}
\end{aligned}$$

□

4.3 Theorem 3

Let p and q be distinct primes and E be a natural number relatively to $(p-1)(q-1)$. Then there exist natural numbers D and y such that

$$ED = 1 + y(p-1)(q-1).$$

Proof. Let p and q be distinct primes and E be a natural number relatively to $(p-1)(q-1)$, thus $(E,(p-1)(q-1)) = 1$. By Theorem 1.40, we can express $(E,(p-1)(q-1)) = 1$ as $Ea+b(p-1)(q-1) = 1, \exists a, b \in \mathbf{Z}$. Now, since Natural numbers fall under integers, we can say that given natural number D and y where $a = D$ and $b = -y$, then;

$$\begin{aligned} Ea + b(p-1)(q-1) &= 1 \\ ED - y(p-1)(q-1) &= 1 \\ ED &= 1 + y(p-1)(q-1) \end{aligned}$$

Thus, there exist natural numbers D and y such that $ED = 1 + y(p-1)(q-1)$. □

4.4 Theorem 4

For all natural numbers n, $(n, n+1) = 1$.

Proof. **Base case (n = 1):**

$$\begin{aligned} (n, n+1) &= 1 \\ (1, 2) &= 1 \end{aligned}$$

Theorem holds for n = 1.

Inductive Hypothesis: Assume n = n + 1, then Theorem still holds.

Inductive Step:

$$
\begin{aligned}
(n, n+1) &= 1 \\
(n+1, n+2) &= 1 \\
(n+1)x + (n+2)y &= 1 \qquad \text{For some integer x and y.} \\
nx + x + ny + 2y &= 1 \\
n(x+y) + 1(x+2y) &= 1 \\
(n, 1) &= 1
\end{aligned}
$$

Therefore, the theorem still holds. □

4.5 Theorem 5 (Infinitude of Primes Theorem)

There are infinitely many prime numbers.

Proof. Let P be a finite set of primes, $P = \{p_1, p_2 ... p_m\}$. We need to show that there will be prime numbers not in P. Now let q be the product of all elements in the set P such that $q = p_1 p_2 ... p_m$. By Theorem 4, we know that for all natural numbers n, (n, n+1) = 1 such that n and n+1 are co primes. Therefore, (q, q+1) = 1, meaning that there is a number x that divides $q + 1$ that is not in the finite prime set P. □