Number Theory and the RSA Encryption Algorithm

Steve Ziskind

June 20, 2023

- In 1978, Rivest, Shamir, and Adleman described an algorithm (RSA) for secure communications.

- In 1978, Rivest, Shamir, and Adleman described an algorithm (RSA) for secure communications.
- The RSA algorithm implemented ideas known as Trap Door encryption and Public Key encryption, which were earlier published by Diffie and Hellman. RSA uses some basic results on number theory.

- In 1978, Rivest, Shamir, and Adleman described an algorithm (RSA) for secure communications.
- The RSA algorithm implemented ideas known as Trap Door encryption and Public Key encryption, which were earlier published by Diffie and Hellman. RSA uses some basic results on number theory.
- This talk will provide the number theory background for RSA, and then describe RSA itself. The number theory is important and interesting all on its own.

- In 1978, Rivest, Shamir, and Adleman described an algorithm (RSA) for secure communications.
- The RSA algorithm implemented ideas known as Trap Door encryption and Public Key encryption, which were earlier published by Diffie and Hellman. RSA uses some basic results on number theory.
- This talk will provide the number theory background for RSA, and then describe RSA itself. The number theory is important and interesting all on its own.
- Some supplemental PDF files listed in the bibliography on the last slide will be found in Larry Susanka's master page.

Greatest Common Divisor

- Given two positive integers, a and b, we denote the Greatest Common Divisor (gcd) of them by (a,b)

Greatest Common Divisor

- Given two positive integers, a and b, we denote the Greatest Common Divisor (gcd) of them by (a,b)
- As the name says, the gcd is the largest number that evenly divides them both.

Greatest Common Divisor

- Given two positive integers, a and b, we denote the Greatest Common Divisor (gcd) of them by (a,b)
- As the name says, the gcd is the largest number that evenly divides them both.
- For example, (12,16)=4. As another example, (4,9)=1

Greatest Common Divisor

- Given two positive integers, a and b, we denote the Greatest Common Divisor (gcd) of them by (a,b)
- As the name says, the gcd is the largest number that evenly divides them both.
- For example, (12,16)=4. As another example, (4,9)=1
- When (a,b)=1 we say that a and b are relatively prime, or co-prime. Note that neither needs to be prime on its own, but they share no common divisors.

Greatest Common Divisor

- Given two positive integers, a and b, we denote the Greatest Common Divisor (gcd) of them by (a,b)
- As the name says, the gcd is the largest number that evenly divides them both.
- For example, (12,16)=4. As another example, (4,9)=1
- When (a,b)=1 we say that a and b are relatively prime, or co-prime. Note that neither needs to be prime on its own, but they share no common divisors.
- We note in passing that the gcd is closely related to the Least Common Multiple of a and b, because $gcd(a, b) * lcm(a, b) = a * b$.

Euclidean Algorithm

- Over 2000 years ago, Euclid's Elements described an efficient method for finding the gcd of two positive integers.

Euclidean Algorithm

- Over 2000 years ago, Euclid's Elements described an efficient method for finding the gcd of two positive integers.
- Suppose $a > b$ are the two numbers. If a is a multiple of b, i.e. b evenly divides into a, (written $b \mid a$), then (a,b)=b.

Euclidean Algorithm

- Over 2000 years ago, Euclid's Elements described an efficient method for finding the gcd of two positive integers.
- Suppose $a > b$ are the two numbers. If a is a multiple of b, i.e. b evenly divides into a, (written $b \mid a$), then (a,b)=b.
- If $b \nmid a$ then write $a = q * b + r$, where q is the quotient and $r < b$ is the remainder when a is divided by b.

Euclidean Algorithm

- Over 2000 years ago, Euclid's Elements described an efficient method for finding the gcd of two positive integers.
- Suppose $a > b$ are the two numbers. If a is a multiple of b, i.e. b evenly divides into a, (written $b \mid a$), then (a,b)=b.
- If $b \nmid a$ then write $a = q * b + r$, where q is the quotient and $r < b$ is the remainder when a is divided by b.
- Now the key: whatever divides a and b must also divide r. Therefore $(a, b) = (b, r)$, and the problem can be carried on with smaller numbers. Eventually we will find (a,b).

Euclidean Algorithm

- As an example, to find (679,161) we calculate:

$$\begin{aligned} 679 &= 4 \times 161 + 35 \\ 161 &= 4 \times 35 + 21 \\ 35 &= 21 + 14 \\ 21 &= 14 + 7 \\ 14 &= 2 \times 7 \end{aligned}$$

In short, (679,161)=7.

Euclidean Algorithm

- As an example, to find (679,161) we calculate:

$$\begin{aligned} 679 &= 4 \times 161 + 35 \\ 161 &= 4 \times 35 + 21 \\ 35 &= 21 + 14 \\ 21 &= 14 + 7 \\ 14 &= 2 \times 7 \end{aligned}$$

  In short, (679,161)=7.

- But we can now run the equations backwards:

$$\begin{aligned} 7 &= 21 - 14 = 21 - (35 - 21) = 2 \times 21 - 35 \\ &= 2 \times (161 - 4 \times 35) - 35 = 2 \times 161 - 9 \times 35 \\ &= 2 \times 161 - 9 \times (679 - 4 \times 161) = 38 \times 161 - 9 \times 679 \end{aligned}$$

Euclidean Algorithm

- As an example, to find (679,161) we calculate:

$$\begin{aligned} 679 &= 4 \times 161 + 35 \\ 161 &= 4 \times 35 + 21 \\ 35 &= 21 + 14 \\ 21 &= 14 + 7 \\ 14 &= 2 \times 7 \end{aligned}$$

  In short, (679,161)=7.

- But we can now run the equations backwards:

$$\begin{aligned} 7 &= 21 - 14 = 21 - (35 - 21) = 2 \times 21 - 35 \\ &= 2 \times (161 - 4 \times 35) - 35 = 2 \times 161 - 9 \times 35 \\ &= 2 \times 161 - 9 \times (679 - 4 \times 161) = 38 \times 161 - 9 \times 679 \end{aligned}$$

- Corollary: For any a and b, there exist integers x and y for which $(a, b) = xa + yb$

Divisibility

- Theorem: If $a \mid bc$ and (a,b)=1, then $a \mid c$

Divisibility

- Theorem: If $a \mid bc$ and (a,b)=1, then $a \mid c$
- Note that a and b being relatively prime is needed. For example, $6 \mid (8 \times 9)$, but it divides neither of them individually.

Divisibility

- Theorem: If $a \mid bc$ and (a,b)=1, then $a \mid c$
- Note that a and b being relatively prime is needed. For example, $6 \mid (8 \times 9)$, but it divides neither of them individually.
- Proof: Using the corollary to the Euclidean Algorithm, there are are integers x and y for which $1 = xa + yb$. Multiplying by c, we find that $c = axc + ybc$. But because $a \mid bc$ there must be some k for which $bc = ka$. Substituting, $c = axc + yka = a \times (xc + yk)$. QED

Divisibility

- Theorem: If $a \mid bc$ and (a,b)=1, then $a \mid c$
- Note that a and b being relatively prime is needed. For example, $6 \mid (8 \times 9)$, but it divides neither of them individually.
- Proof: Using the corollary to the Euclidean Algorithm, there are are integers x and y for which $1 = xa + yb$. Multiplying by c, we find that $c = axc + ybc$. But because $a \mid bc$ there must be some k for which $bc = ka$. Substituting, $c = axc + yka = a \times (xc + yk)$. QED
- As a side note, this theorem can be used to give an easy proof of the Fundamental Theorem of Arithmetic, which states that there is only one way to factor an integer into a product of primes. See the bibliography.

Modular Equivalence

- A special notation, originated by Gauss, is used to denote a certain type of equivalence relation. We write $a \equiv b \pmod{m}$ when $m \mid (a - b)$.

Modular Equivalence

- A special notation, originated by Gauss, is used to denote a certain type of equivalence relation. We write $a \equiv b \pmod{m}$ when $m \mid (a - b)$.
- A familiar case occurs with clocks. Every time the hour hand goes around once, the time is the same, except for a multiple of 12. So when counting hours we can say:

$$3 \equiv 15 \equiv 27 \pmod{12}$$

Modular Equivalence

- A special notation, originated by Gauss, is used to denote a certain type of equivalence relation. We write $a \equiv b \pmod{m}$ when $m \mid (a - b)$.
- A familiar case occurs with clocks. Every time the hour hand goes around once, the time is the same, except for a multiple of 12. So when counting hours we can say:

$$3 \equiv 15 \equiv 27 \pmod{12}$$

- With one important exception, equivalent numbers can be treated like ordinary equality. For example, if $a \equiv b \pmod{m}$ and $x \equiv y \pmod{m}$, then $a + x \equiv b + y \pmod{m}$ and $ax \equiv by \pmod{m}$, etc.

Modular Equivalence

- A special notation, originated by Gauss, is used to denote a certain type of equivalence relation. We write $a \equiv b \pmod{m}$ when $m \mid (a - b)$.
- A familiar case occurs with clocks. Every time the hour hand goes around once, the time is the same, except for a multiple of 12. So when counting hours we can say:

$$3 \equiv 15 \equiv 27 \pmod{12}$$

- With one important exception, equivalent numbers can be treated like ordinary equality. For example, if $a \equiv b \pmod{m}$ and $x \equiv y \pmod{m}$, then $a + x \equiv b + y \pmod{m}$ and $ax \equiv by \pmod{m}$, etc.
- The exception is division. $4 \times 5 \equiv 4 \times 2 \pmod{6}$, but we cannot divide both sides by 4, because 5 and 2 are not equivalent.

Fermat's Observation

- Let us begin with an arbitrary prime number: 7. List all the positive numbers less than 7: $\{1,2,3,4,5,6\}$.

Fermat's Observation

- Let us begin with an arbitrary prime number: 7. List all the positive numbers less than 7: $\{1,2,3,4,5,6\}$.
- Multiply each of the numbers by 3, and find the equivalent smallest number $\pmod 7$:

$$
\begin{aligned}
3 \times 1 &= 3 \\
3 \times 2 &= 6 \\
3 \times 3 &= 9 \equiv 2 \\
3 \times 4 &= 12 \equiv 5 \\
3 \times 5 &= 15 \equiv 1 \\
3 \times 6 &= 18 \equiv 4
\end{aligned}
$$

Fermat's Observation

- Let us begin with an arbitrary prime number: 7. List all the positive numbers less than 7: $\{1,2,3,4,5,6\}$.
- Multiply each of the numbers by 3, and find the equivalent smallest number $\pmod 7$:

$$3 \times 1 = 3$$
$$3 \times 2 = 6$$
$$3 \times 3 = 9 \equiv 2$$
$$3 \times 4 = 12 \equiv 5$$
$$3 \times 5 = 15 \equiv 1$$
$$3 \times 6 = 18 \equiv 4$$

- We have the exact same 6 numbers, just in a different order. This was not an accident!

Fermat's Little Theorem

- Fermat's Little Theorem: Let p be a prime number, and let (a,p)=1. Then
$$a^{p-1} \equiv 1 \pmod{p}$$

Fermat's Little Theorem

- Fermat's Little Theorem: Let p be a prime number, and let (a,p)=1. Then
$$a^{p-1} \equiv 1 \pmod{p}$$
- Proof: There are $p-1$ positive numbers, $\{1, 2, \ldots, p-1\} = \{x_k\}$ each relatively prime to p. Multiply each by a and reduce to something less than p. If $ax_i \equiv ax_j$ then $p \mid a \times (x_i - x_j)$. But $p \nmid a$, so our previous theorem says that $p \mid (x_i - x_j)$, which is clearly impossible. Therefore the set of numbers $\{ax_i\} \equiv \{x_i\} \pmod{p}$

Fermat's Little Theorem

- Fermat's Little Theorem: Let p be a prime number, and let (a,p)=1. Then
$$a^{p-1} \equiv 1 \pmod{p}$$
- Proof: There are $p-1$ positive numbers, $\{1, 2, \ldots, p-1\} = \{x_k\}$ each relatively prime to p. Multiply each by a and reduce to something less than p. If $ax_i \equiv ax_j$ then $p \mid a \times (x_i - x_j)$. But $p \nmid a$, so our previous theorem says that $p \mid (x_i - x_j)$, which is clearly impossible. Therefore the set of numbers $\{ax_i\} \equiv \{x_i\} \pmod{p}$
- Now we know that

$$\prod x_i \equiv \prod ax_i = a^{p-1} \prod x_i \pmod{p}$$

Fermat's Little Theorem

- Fermat's Little Theorem: Let p be a prime number, and let (a,p)=1. Then
$$a^{p-1} \equiv 1 \pmod{p}$$
- Proof: There are $p-1$ positive numbers, $\{1, 2, \ldots, p-1\} = \{x_k\}$ each relatively prime to p. Multiply each by a and reduce to something less than p. If $ax_i \equiv ax_j$ then $p \mid a \times (x_i - x_j)$. But $p \nmid a$, so our previous theorem says that $p \mid (x_i - x_j)$, which is clearly impossible. Therefore the set of numbers $\{ax_i\} \equiv \{x_i\} \pmod{p}$
- Now we know that
$$\prod x_i \equiv \prod ax_i = a^{p-1} \prod x_i \pmod{p}$$
- So $p \mid (a^{p-1} - 1) \prod x_i$. But $p \nmid \prod x_i$, so our previous theorem ends the proof. QED

Euler's Extension

- Roughly 100 years after Fermat, Euler noticed that the key to the proof of Fermat's Little Theorem was that the numbers less than p were all relative prime to it. The $p - 1$ in the exponent was just the number of such relative prime numbers.

Euler's Extension

- Roughly 100 years after Fermat, Euler noticed that the key to the proof of Fermat's Little Theorem was that the numbers less than p were all relative prime to it. The $p - 1$ in the exponent was just the number of such relative prime numbers.
- Euler defined a function, $\phi(n)$ as the number of numbers less than n that are relatively prime to n. For example, if n $=$ 15, then the numbers less than 15 relatively prime to it are $\{1, 2, 4, 7, 8, 11, 13, 14\}$, so $\phi(15) = 8$. This function is sometimes called the totient.

Euler's Extension

- Roughly 100 years after Fermat, Euler noticed that the key to the proof of Fermat's Little Theorem was that the numbers less than p were all relative prime to it. The $p-1$ in the exponent was just the number of such relative prime numbers.
- Euler defined a function, $\phi(n)$ as the number of numbers less than n that are relatively prime to n. For example, if n = 15, then the numbers less than 15 relatively prime to it are $\{1, 2, 4, 7, 8, 11, 13, 14\}$, so $\phi(15) = 8$. This function is sometimes called the totient.
- Euler's Theorem: Let any two numbers satisfy (a,n)=1. Then

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

Euler's Extension

- Roughly 100 years after Fermat, Euler noticed that the key to the proof of Fermat's Little Theorem was that the numbers less than p were all relative prime to it. The $p - 1$ in the exponent was just the number of such relative prime numbers.
- Euler defined a function, $\phi(n)$ as the number of numbers less than n that are relatively prime to n. For example, if n = 15, then the numbers less than 15 relatively prime to it are $\{1, 2, 4, 7, 8, 11, 13, 14\}$, so $\phi(15) = 8$. This function is sometimes called the totient.
- Euler's Theorem: Let any two numbers satisfy (a,n)=1. Then

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

- The proof is almost word for word identical to that for Fermat's Theorem.

Using Euler

- Theorem: If (a,m)=1 then $ax \equiv b \pmod{m}$ has a solution.

Using Euler

- <u>Theorem:</u> If (a,m)=1 then $ax \equiv b \pmod{m}$ has a solution.
- <u>Proof:</u> Let $x = a^{\phi(m)-1}b$ and apply Euler. QED

Using Euler

- Theorem: If (a,m)=1 then $ax \equiv b \pmod{m}$ has a solution.
- Proof: Let $x = a^{\phi(m)-1}b$ and apply Euler. QED
- Chinese Remainder Theorem: Let $m_1, m_2, \ldots, m_n$ be pairwise relatively prime, and let $b_1, \ldots, b_n$ be arbitrary. Then there is a number x that simultaneously solves $x \equiv b_i \pmod{m_i}$

Using Euler

- Theorem: If (a,m)=1 then $ax \equiv b \pmod{m}$ has a solution.
- Proof: Let $x = a^{\phi(m)-1}b$ and apply Euler. QED
- Chinese Remainder Theorem: Let $m_1, m_2, \ldots, m_n$ be pairwise relatively prime, and let $b_1, \ldots, b_n$ be arbitrary. Then there is a number x that simultaneously solves $x \equiv b_i \pmod{m_i}$
- Proof (n=3): Similar in spirit to Lagrange Interpolation, use the above result to solve the three equations:

$$\begin{aligned} m_2 m_3 x_1 &\equiv b_1 \pmod{m_1} \\ m_1 m_3 x_2 &\equiv b_2 \pmod{m_2} \\ m_1 m_2 x_3 &\equiv b_3 \pmod{m_3} \end{aligned}$$

Now let $x = m_2 m_3 x_1 + m_1 m_3 x_2 + m_1 m_2 x_3$. QED

The Product Formula for ϕ

- Theorem: If (X,Y)=1 then $\phi(XY) = \phi(X)\phi(Y)$

The Product Formula for ϕ

- Theorem: If (X,Y)=1 then $\phi(XY) = \phi(X)\phi(Y)$
- Proof: Let $A = \{a_1, \ldots, a_m\}$ be the distinct coprime residues of X, likewise $= \{b_1, \ldots, b_n\}$ for Y and $C = \{c_1, \ldots\}$ for XY. Define a mapping on C by $T(c) =< a, b >$, where $c \equiv a \pmod{X}$ and $c \equiv b \pmod{Y}$. We need to show that T is a 1-to-1 mapping of C onto $A \times B$.

The Product Formula for ϕ

- Theorem: If (X,Y)=1 then $\phi(XY) = \phi(X)\phi(Y)$
- Proof: Let $A = \{a_1, \ldots, a_m\}$ be the distinct coprime residues of X, likewise $= \{b_1, \ldots, b_n\}$ for Y and $C = \{c_1, \ldots\}$ for XY. Define a mapping on C by $T(c) =< a, b >$, where $c \equiv a \pmod{X}$ and $c \equiv b \pmod{Y}$. We need to show that T is a 1-to-1 mapping of C onto $A \times B$.
- Claim 1: (a,X)=1 and (b,Y)=1. For if not then there is some $\lambda > 1$ for which $\lambda \mid a$ and $\lambda \mid X$. But $c = a + \mu X$ which would mean that $\lambda \mid c$. Now we have both $\lambda \mid c$ and $\lambda \mid XY$, violating (c,XY)=1. Similarly for (b,Y).

The Product Formula for ϕ

- Theorem: If (X,Y)=1 then $\phi(XY) = \phi(X)\phi(Y)$
- Proof: Let $A = \{a_1, \ldots, a_m\}$ be the distinct coprime residues of X, likewise $= \{b_1, \ldots, b_n\}$ for Y and $C = \{c_1, \ldots\}$ for XY. Define a mapping on C by $T(c) =< a, b >$, where $c \equiv a \pmod{X}$ and $c \equiv b \pmod{Y}$. We need to show that T is a 1-to-1 mapping of C onto $A \times B$.
- Claim 1: (a,X)=1 and (b,Y)=1. For if not then there is some $\lambda > 1$ for which $\lambda \mid a$ and $\lambda \mid X$. But $c = a + \mu X$ which would mean that $\lambda \mid c$. Now we have both $\lambda \mid c$ and $\lambda \mid XY$, violating (c,XY)=1. Similarly for (b,Y).
- Claim 2: Distinct choices of c yield distinct $< a, b >$ pairs. For if $T(c_1) = T(c_2) =< a, b >$, then $c_1 \equiv a \equiv c_2 \pmod{X}$ and $c_1 \equiv a \equiv c_2 \pmod{Y}$. Denoting $d = c_1 - c_2$ we have, for some λ and μ, $d = \lambda X = \mu Y$. But for some p and q, $1 = pX + pqY$, so $d = pdX + qdY = p\mu YX + q\lambda XY$. Thus d is a multiple of XY, and so $c_1 \equiv c_2 \pmod{XY}$. So the c choices were really the same.

The Product Formula for ϕ

- Claim 3: Every pair $< a_i, b_j >$ arises as $T(c_k)$. By the Chinese Remainder Theorem, there is some c for which $T(c) =< a_i, b_j >$. We need to show that (c,XY)=1. Note first that (c,X)=1 because any divisor of c and X would also divide a. Likewise, (c,Y)=1. But if something divided both c and XY, and it cannot divide X, it would force it to divide Y. This is impossible.

The Product Formula for ϕ

- Claim 3: Every pair $< a_i, b_j >$ arises as $T(c_k)$. By the Chinese Remainder Theorem, there is some c for which $T(c) =< a_i, b_j >$. We need to show that (c,XY)=1. Note first that (c,X)=1 because any divisor of c and X would also divide a. Likewise, (c,Y)=1. But if something divided both c and XY, and it cannot divide X, it would force it to divide Y. This is impossible.
- The 3 above claims show that T establishes a 1-1 correspondence between members of C and the product set $A \times B$. QED

The Product Formula for ϕ

- Claim 3: Every pair $< a_i, b_j >$ arises as $T(c_k)$. By the Chinese Remainder Theorem, there is some c for which $T(c) =< a_i, b_j >$. We need to show that (c,XY)=1. Note first that (c,X)=1 because any divisor of c and X would also divide a. Likewise, (c,Y)=1. But if something divided both c and XY, and it cannot divide X, it would force it to divide Y. This is impossible.
- The 3 above claims show that T establishes a 1-1 correspondence between members of C and the product set $A \times B$. QED
- Corollary: If p and q are distinct primes then $\phi(pq) = (p-1)(q-1)$

The Product Formula for ϕ

- Claim 3: Every pair $< a_i, b_j >$ arises as $T(c_k)$. By the Chinese Remainder Theorem, there is some c for which $T(c) =< a_i, b_j >$. We need to show that (c,XY)=1. Note first that (c,X)=1 because any divisor of c and X would also divide a. Likewise, (c,Y)=1. But if something divided both c and XY, and it cannot divide X, it would force it to divide Y. This is impossible.
- The 3 above claims show that T establishes a 1-1 correspondence between members of C and the product set $A \times B$. QED
- Corollary: If p and q are distinct primes then $\phi(pq) = (p-1)(q-1)$
- More generally, if $n = \prod p_i^{e_i}$ then $\phi(n) = n \prod (1 - \frac{1}{p_i})$

Encryption Goals

- As we all know, when messages are transmitted via computers they are turned into streams of bits, essentially long strings of zeros and ones.

Encryption Goals

- As we all know, when messages are transmitted via computers they are turned into streams of bits, essentially long strings of zeros and ones.
- A very long string can be broken into blocks/packets, that are sent separately and reassembled. Each block of bits is effectively a number in base 2.

Encryption Goals

- As we all know, when messages are transmitted via computers they are turned into streams of bits, essentially long strings of zeros and ones.
- A very long string can be broken into blocks/packets, that are sent separately and reassembled. Each block of bits is effectively a number in base 2.
- The central goal of encryption is to prevent someone from reading what they should not, but because messages can be intercepted we want them to be unintelligible to any unintended recipient.

Encryption Goals

- As we all know, when messages are transmitted via computers they are turned into streams of bits, essentially long strings of zeros and ones.
- A very long string can be broken into blocks/packets, that are sent separately and reassembled. Each block of bits is effectively a number in base 2.
- The central goal of encryption is to prevent someone from reading what they should not, but because messages can be intercepted we want them to be unintelligible to any unintended recipient.
- Each transmitted block, which is a number, can be considered a message (M). RSA aims to turn M into another number so that the intended receiver, and only that receiver, can recover M.

Encryption Goals

- As we all know, when messages are transmitted via computers they are turned into streams of bits, essentially long strings of zeros and ones.
- A very long string can be broken into blocks/packets, that are sent separately and reassembled. Each block of bits is effectively a number in base 2.
- The central goal of encryption is to prevent someone from reading what they should not, but because messages can be intercepted we want them to be unintelligible to any unintended recipient.
- Each transmitted block, which is a number, can be considered a message (M). RSA aims to turn M into another number so that the intended receiver, and only that receiver, can recover M.
- The method has other nice properties.

RSA Mechanization

- RSA is a Public Key method. Every user has a personal pair of keys, one for encrypting and the other for decrypting. Each key is a pair of numbers: (e,n) and (d,n).

RSA Mechanization

- RSA is a Public Key method. Every user has a personal pair of keys, one for encrypting and the other for decrypting. Each key is a pair of numbers: (e,n) and (d,n).
- The encryption key, (e,n), is published widely for all to see and use. The decryption key, (d,n), is tightly and privately held. The user also knows, and tightly protects, the Euler Phi function of n, $\phi(n)$.

RSA Mechanization

- RSA is a Public Key method. Every user has a personal pair of keys, one for encrypting and the other for decrypting. Each key is a pair of numbers: (e,n) and (d,n).
- The encryption key, (e,n), is published widely for all to see and use. The decryption key, (d,n), is tightly and privately held. The user also knows, and tightly protects, the Euler Phi function of n, $\phi(n)$.
- The number n is chosen as the product of two distinct, very large primes, p and q. They should be hundreds of digits long. This makes $\phi(n) = (p-1)(q-1)$

RSA Mechanization

- RSA is a Public Key method. Every user has a personal pair of keys, one for encrypting and the other for decrypting. Each key is a pair of numbers: (e,n) and (d,n).
- The encryption key, (e,n), is published widely for all to see and use. The decryption key, (d,n), is tightly and privately held. The user also knows, and tightly protects, the Euler Phi function of n, $\phi(n)$.
- The number n is chosen as the product of two distinct, very large primes, p and q. They should be hundreds of digits long. This makes $\phi(n) = (p-1)(q-1)$
- Next, each user finds another large number e that is relatively prime to $\phi(n)$. Another large (prime) number will probably work well, because $\phi(n)$ is nearly equal to n.

RSA Mechanization

- RSA is a Public Key method. Every user has a personal pair of keys, one for encrypting and the other for decrypting. Each key is a pair of numbers: (e,n) and (d,n).
- The encryption key, (e,n), is published widely for all to see and use. The decryption key, (d,n), is tightly and privately held. The user also knows, and tightly protects, the Euler Phi function of n, $\phi(n)$.
- The number n is chosen as the product of two distinct, very large primes, p and q. They should be hundreds of digits long. This makes $\phi(n) = (p-1)(q-1)$
- Next, each user finds another large number e that is relatively prime to $\phi(n)$. Another large (prime) number will probably work well, because $\phi(n)$ is nearly equal to n.
- Finally, each user determines d by solving $de \equiv 1 \pmod{\phi(n)}$. This needs to be secret.

RSA Mechanization

- To send a secure message (M) to someone, the sender first looks up the receivers encryption key, (e,n), and calculates $E(M) \equiv M^e \pmod{n}$.

RSA Mechanization

- To send a secure message (M) to someone, the sender first looks up the receivers encryption key, (e,n), and calculates $E(M) \equiv M^e \pmod{n}$.
- The whole world can safely see E(M) without knowing M.

RSA Mechanization

- To send a secure message (M) to someone, the sender first looks up the receivers encryption key, (e,n), and calculates $E(M) \equiv M^e \pmod{n}$.
- The whole world can safely see E(M) without knowing M.
- When E(M) is received, the recipient calculates $D(E(M)) \equiv (EM)^d \pmod{n}$.

RSA Mechanization

- To send a secure message (M) to someone, the sender first looks up the receivers encryption key, (e,n), and calculates $E(M) \equiv M^e \pmod{n}$.
- The whole world can safely see E(M) without knowing M.
- When E(M) is received, the recipient calculates $D(E(M)) \equiv (EM)^d \pmod{n}$.
- Because $de \equiv 1 \pmod{\phi(n)}$, $de = k\phi(n) + 1$ for some k.

RSA Mechanization

- To send a secure message (M) to someone, the sender first looks up the receivers encryption key, (e,n), and calculates $E(M) \equiv M^e \pmod{n}$.
- The whole world can safely see E(M) without knowing M.
- When E(M) is received, the recipient calculates $D(E(M)) \equiv (EM)^d \pmod{n}$.
- Because $de \equiv 1 \pmod{\phi(n)}$, $de = k\phi(n) + 1$ for some k.
- This means that $D(E(M)) = M^{de} = M^{k\phi(n)+1} = (M^{\phi(n)})^k \times M$ $\equiv 1^k \times M = M$.

RSA Mechanization

- To send a secure message (M) to someone, the sender first looks up the receivers encryption key, (e,n), and calculates $E(M) \equiv M^e \pmod{n}$.
- The whole world can safely see E(M) without knowing M.
- When E(M) is received, the recipient calculates $D(E(M)) \equiv (EM)^d \pmod{n}$.
- Because $de \equiv 1 \pmod{\phi(n)}$, $de = k\phi(n) + 1$ for some k.
- This means that $D(E(M)) = M^{de} = M^{k\phi(n)+1} = (M^{\phi(n)})^k \times M$ $\equiv 1^k \times M = M$.
- The original message has been recovered!

Signatures

- There is an important and slightly subtle problem. Anyone with access to a person's public encryption key can use it to send a secure message, but how will the receiver know who sent it?

Signatures

- There is an important and slightly subtle problem. Anyone with access to a person's public encryption key can use it to send a secure message, but how will the receiver know who sent it?
- A neat trick solves this problem. The sender appends to his message a signature message that he has encrypted with his decryption key. The recipient uses his decryption key to unscramble the body of the message, which tells him who the sender is. At the end of the message is a scrambled number that can be unlocked with the sender's public encryption key. Anyone trying to impersonate a sender would not be able to build something that would be unlocked by the purported sender's encryption key. Spoofing and impersonation are prevented.

A few caveats

- When M^e is reduced mod n, the result will be a number smaller than n. This means that n should be large, otherwise M will be forced to be small. Typically, p and q are chosen to be several hundred digits long.

A few caveats

- When M^e is reduced mod n, the result will be a number smaller than n. This means that n should be large, otherwise M will be forced to be small. Typically, p and q are chosen to be several hundred digits long.
- Finding a random 200 digit prime number is not entirely trivial, but the Prime Number Theorem assures us that a random 200 digit number has probability of 0.72% of being prime. So generating several hundred random such numbers is very likely to have a prime in the list. We just need to find it in the list. Most can be instantly eliminated (e.g. they are even or end in 5), and there are both simple and sophisticated tests that will (probably) eliminate any composites.

A few caveats

- When M^e is reduced mod n, the result will be a number smaller than n. This means that n should be large, otherwise M will be forced to be small. Typically, p and q are chosen to be several hundred digits long.
- Finding a random 200 digit prime number is not entirely trivial, but the Prime Number Theorem assures us that a random 200 digit number has probability of 0.72% of being prime. So generating several hundred random such numbers is very likely to have a prime in the list. We just need to find it in the list. Most can be instantly eliminated (e.g. they are even or end in 5), and there are both simple and sophisticated tests that will (probably) eliminate any composites.
- For Euler's theorem to hold, M must be relatively prime to n. $\phi(n)$ is very close to n, so the chance of a common factor is exceedingly small. You can just take the chance or, if you are the jittery type, run them through the Euclidean Algorithm at a slight cost of processing time.

Bibliography

- “Euclidean Algorithm”
- “The Fundamental Theorem of Arithmetic”
- Larry Susanka, “Number Theory”
- Evgeny Milanov, “The RSA Algorithm”