Euler's Totient Function and Public Key Cryptography

Clay S. Turner
Nov 7, 2008
mailto:clay@claysturner.com

Introduction

Several times as a consultant, I've had the opportunity to design and implement some cryptosystems. Invariably I need to explain to the client the theory behind the methods and this discussion leads to totients. Since many have never heard of or are familiar with Euler's totient function, I thought I'd put together a paper describing this function and its relation to public key cryptography.

I will keep this paper in a somewhat informal style, but I will use some seemingly arcane mathematics terms. However when I use them I will provide their definitions.

Euler's Totient Function.

In order to talk about this, I will need to give a few definitions – some of these should already be familiar to the reader. So here goes:

- Natural Number – a number from the set of $\{1,2,3,\cdots,\infty\}$
- Integer – a number in the set $\{-\infty,\cdots,-2,-1,0,1,2,\cdots,\infty\}$
- Prime Number – a natural number (greater than one) that is only divisible by one and itself.
- Composite Number – a natural number (greater than one) that is not prime.
- Greatest Common Divisor – the largest common factor of a set of numbers.
- $\gcd(a,b)$ - mathematical shorthand for the greatest common divisor of a and b.
- Coprime – a set of numbers is coprime if their greatest common divisor is one.
- Relatively Prime – the same thing as coprime.
- Totative – a number, $m < n$, is a totative of n if $\gcd(m,n) = 1$, where n is a natural number.

So now with these definitions we can quite tersely define Euler's[1] totient function, $\varphi(n)$, as the number of totatives of n. Sometimes the Euler totient function is called Euler's phi function or simply the phi function. What does this definition really mean? It means that the Euler totient function gives a count of how many numbers in the set, $\{1,2,3,\cdots,n\}$

[1] Leonhard Paul Euler [1707 - 1783], a Swiss mathematician and physicist, who made a great number of contributions to the fields of Calculus, Graph Theory, Mathematical Analysis, Mechanics, Optics, and Astronomy. He is considered by many to be the top mathematician of the 18th century.

share no common factors with n that are greater than one. We will soon get in to evaluating the totient.

Brief History

Euler's name is attached to this function since he invented[2] it during the early to mid 1700s and used it to prove Fermat's[3] little theorem and derived from it his own more general theorem. Euler never used the term "totient" as that was coined over a century later by Sylvester[4] in 1879 [2].

While the term *totient* sometimes refers to a convolution[5] product of two multiplicative sequences, the term (in that way) is now really only used with two such sequences. Namely the Euler and Jordan. The exact reasoning that Sylvester used to coin the word is not known, but it appears to be a cross between "total" and "quotient." Note Euler's totient function is multiplicative and not completely multiplicative. I'll cover what multiplicative means under the section dealing with the properties of the totient function.

Evaluation

To derive a formula for evaluating, $\varphi(n)$, we will apply the fundamental theorem of arithmetic[6] which says that every natural number greater than one has a unique factorization in terms of prime numbers. I.e., $n = p_1^{k_1} p_2^{k_2} \cdots p_m^{k_m}$, where p_m are the m distinct prime factors of n. For example, if we have, $n = 30$, its prime factor representation is $2 \times 3 \times 5$. Likewise for $n = 72$, we find it factors into $8 \times 9 = 2^3 \cdot 3^2$.

For n prime, it follows that all positive integers less then n are coprime with n, thus $\phi(n) = n - 1$. But for, n, composite we will have to do a little more work. To make this development a little easier to follow, we will work with a specific example, namely for $n = 30$. First let's define the set, S, to be $\{1,2,\cdots,29,30\}$. Then we recall that $\varphi(30)$ is simply the count of natural numbers in the set S that are coprime with the number 30.

A naïve way to evaluate the totient is had by writing down each of the members of the set S, and then striking out all of those who share a common factor greater than 1 with *n*. For n=30, we see that 2 is a factor, so we strike out 2 and all of its multiples. Likewise 3 is a factor of 30, so again we strike it out and all of its multiples. The same thing happens with 5. But 7 is not a factor, so we leave the number 7 alone. We can test all of the unstruck members this way. But in fact we only have to test all primes less than or equal to the square root of n. So for the number 30, we only need to test 2, 3, and 5. After all of

---

[2] Euler's proof was published in 1736

[3] Pierre de Fermat [1601 – 1665], a French lawyer and mathematician who made many of the developments which led to the invention of the calculus. He is also known for his work in number theory.

[4] James Joseph Sylvester [1814 - 1897], A British mathematician who made fundamental contributions to matrix theory, number theory, partition theory, and combinatorics. He was also known for naming many mathematical things. He most well known term is likely to be the word, *discriminant*, usually encountered when one 1st learns to use the general quadratic formula.

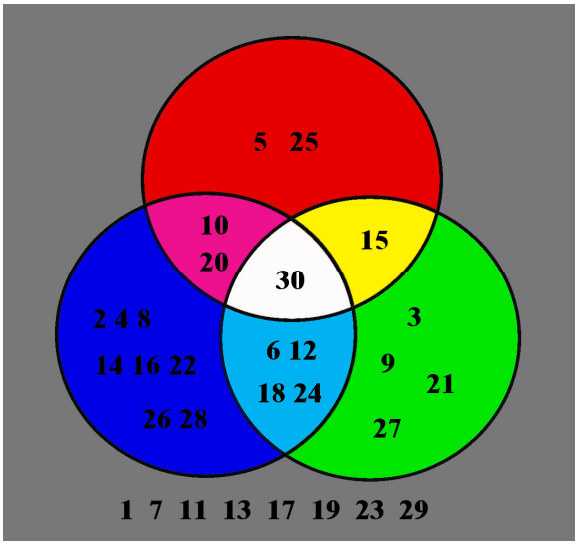
[5] In this case the convolution is a Dirichlet Convolution. See Appendix B for details.

[6] This theorem dates back to at least 300BC where Euclid published a proof in his book, "Elements."

the striking out is completed, then the totient is simply the count of the unstruck members. This is akin (although a little different) to the Sieve of Eratosthenes[7] used to find prime numbers.

Now we will look at another, more useful, way that exploits Euclid's theorem about prime factors. So for our specific example of 30 let's define three new sets – one for each prime factor. Let A, be the set of integers from S that 2 divides into evenly. Let B, be the set of integers from S that 3 divides into evenly, and finally let C, be the set of integers from S that 5 divides into S evenly.

The following Venn[8] diagram shows the three sets and their overlaps. The blue region has the multiples of 2, the green region has multiples of 3, and the red region has multiples of 5. The magenta, cyan, yellow, and white regions are the overlap regions where the numbers have two or more prime factors in common with the factors of 30. The grey region is the set of totatives of 30.



Now what we want is the size of the set of natural numbers in S that are coprime with the number 30. Hence we want: $\phi(30) = sizeof(\overline{A \cup B \cup C})$, where the over bar means complement. Of course the three sets have some overlap, so we can't just count the number of members in each and tally the results without accounting for the overlap. But we can apply De Morgan's[9] theorem for sets and get the following statement.

$$\varphi(30) = sizeof(\overline{A} \cap \overline{B} \cap \overline{C})$$

---

[7] Eratosthenes of Cyrene [276BC – 194BC], was a Greek mathematician, geographer, astronomer, and poet. He was the 1st to calculate a reasonable circumference for the Earth. His sieve method to find prime numbers is now commonly used by computers as one means to evaluate a computer's performance.

[8] John Venn [1834 - 1923] was a British logician and philosopher who gave us the frequency interpretation of probability. His book on symbolic logic gave us his eponymous diagrams.

[9] Augustus De Morgan [1806 – 1871] A British Logician and Mathematician who not only formulated his laws concerning set theory, but he also gave us the term mathematical induction and made it rigorous.

Now we can use John Venn's idea about the frequency interpretation of probability. For set *A*, (which contains 2 and all of its multiples), the probability that a number from set *S* chosen at random is in set *A* is simply $1/2$. Likewise the probability that a number from *S* is in *B*, is $1/3$. And for set *C* we have $1/5$. Of course the probability of being outside a set is simply one minus the probability of being inside the set. Taking Venn's idea further, we see that the probability of being outside the intersection of two sets is the product of the individual probabilities of being outside of either set alone. So the probability of a number in set *S* being outside of sets *A*, *B*, and *C* simultaneously is:

$$P_{outside} = (1-1/2)(1-1/3)(1-1/5)$$

And finally reversing the frequency-probability interpretation (going from probability back to frequency) we have for our example:

$$\varphi(30) = 30\left(1-\frac{1}{2}\right)\left(1-\frac{1}{3}\right)\left(1-\frac{1}{5}\right)$$

which yields $\varphi(30) = 8$. We can now write a general formula for Euler's totient in terms of prime factors.

$$\varphi(n) = n\left(1-\frac{1}{p_1}\right)\left(1-\frac{1}{p_2}\right)\cdots\left(1-\frac{1}{p_m}\right)$$

Here the p_m are the set of m prime factors of n. Note: with this formula the *m* prime factors are the distinct factors of *n*. For example, $\varphi(256) = \varphi(2^8) = 256(1-1/2) = 128$, and $\varphi(72) = \phi\left(2^3 \cdot 3^2\right) = 72(1-1/2)(1-1/3) = 72(1/3) = 24$.

Using the prime number factorization for *n,* we can find another formula for $\phi(n)$. A little manipulation (replacing *n* with its prime factor representation and grouping like terms), we find

$$\phi(n) = (p_1-1)p_1^{k_1-1}(p_2-1)p_2^{k_2-1}\cdots(p_m-1)p_m^{k_m-1}$$

In this case the multiplicity of each factor is required. For example, $\varphi(72) = \varphi(2^3 \cdot 3^2) = (2-1)2^{3-1} \cdot (3-1)3^{2-1} = 1 \cdot 2^2 \cdot 2 \cdot 3^1 = 24$.

General Properties

We can immediately see that the number one is a totative for any natural number, so $\varphi(n) \geq 1$ for all natural *n*. Also we see that the number n is itself not a totative of n, so also have an upper bound for phi. Thus,

$$1 \le \varphi(n) \le n-1$$

It turns out that the only time $\varphi(n) = n-1$ is when n is prime.

If we have two coprime numbers, n and m, we can easily (by the 2nd evaluation formula) arrive at the relation $\varphi(n \times m) = \varphi(n) \times \varphi(m)$. This and the fact that $\varphi(1) = 1$, makes the totient function multiplicative. For a function to be completely multiplicative, the factoring can't have any restrictions such as the coprime one for Euler's totient.

Fermat's Little Theorem[10]

Fermat, in 1640, disclosed in a letter a theorem without proof (claiming the proof would be too long) that stated for any integer a and prime p that

$$a^p \equiv a \pmod{p}$$

Now if a and p are coprime we can rewrite this as

$$a^{p-1} \equiv 1 \pmod{p}$$

The coprime requirement comes from dividing out the radix. The first published proof of FLT is due to Euler in 1736. However, it has been discovered that Leibniz[11] had a similar proof in an unpublished manuscript that predated 1683. A simple proof is presented in Appendix A.

Euler's Generalization of Fermat's Theorem

Euler extended the notion of Fermat's Little theorem so that if a and n are coprime and n is any modulus, then

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

Which we can see contains Fermat's little theorem as a special case.

RSA Algorithm

In 1976 Diffie and Hellman [3] published a landmark paper describing properties of public key cryptosystems utilizing "one-way" trap door functions. Rivest, Shamir, and

[10] Do not confuse this theorem with Fermat's Last Theorem which was finally proven (1994) by British mathematician Andrew Wiles [1953 -].

[11] Gottfried Leibniz [1646 – 1716] A German Mathematician and Philosopher who developed Calculus independently of Newton. In fact it is the Leibniz notation that is in common use today.

Adleman knew of a trap door function and devised a clever public key system around it. This system, now known as RSA, is widely used. And Euler's totient function is at the heart of the method.

The RSA method uses the same mathematical form (modular exponentiation) for both encryption and decryption. I.e. given a plaintext (message), M, represented as a number, then the cyphertext, C, is found by:

$$C = M^e \pmod{n}$$

Likewise the decryption is found by:

$$M = C^d \pmod{n}$$

The public key is the pair $\{e, n\}$ and the private key is the pair $\{d, n\}$. The security stems from the linkage between *d* and *e* being through the prime factors of *n*. If n is made up as the product of two very large prime numbers, then the difficulty of factoring n is what stops one from finding *d* when given *e* and *n*. The RSA method exploits the fact that factoring a large number into smaller components is more difficult than forming a large number by multiplying together smaller numbers. If the large number is a product of two near equal length primes, then this computational disparity can be quite extreme. This is an example of a one way trap door function. To ensure the one wayness of this function, RSA type cryptosystems use numbers with thousands of digits.

To create keys for RSA, one does the following steps:

1. Picks (randomly) two large prime numbers and calls them *p* and *q*.
2. Calculates their product and calls it *n*.
3. Calculates the totient of n ; it is simply $(p-1)(q-1)$.
4. Picks a random integer that is coprime to $\phi(n)$ and calls this *e*. A simple way is to just pick a random number $> \max(p, q)$.
5. Calculates (via the Euclidean division algorithm) the multiplicative inverse of e modulo $\phi(n)$ and call this number *d*.

Now one has both their public and private keys. Rivest et al, in [1] give algorithms for efficient implementation of these various steps. Quite a bit has been written about the security offered by RSA and if a few rules are followed, the encryption is practically unbreakable. In fact most successful attacks utilize side channel information such as timing variations, radio emissions, power consumption variations, etc and looking for poorly chosen primes, i.e., Mersenne and Fermat primes.

Now let's look at why this method works. First we start with Euler's generalization of Fermat's little theorem. It is:

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

This is true if M and *n* are coprime. Since the LHS is congruent to one, we can raise it to any integral power, *k*, without changing its value, thus:

$$M^{k \cdot \phi(n)} \equiv 1 \pmod{n}$$

Finally we will multiply both sides by M, yielding:

$$M^{k \cdot \phi(n)+1} \equiv M \pmod{n}$$

Hence we see that modulator exponentiation to a power of $k \cdot \phi(n)+1$ is an identity operation. We will now see that the RSA method is a way of splitting this identity operation into two steps – one for encryption and the other for decryption. The multiplication by M also removes the coprime requirement between M and *n*. It is assumed that $0 \le M < n$.

Since we require that any message encrypted and then decrypted be the original message, then we have:

$$\left(M^{e}\right)^{d} = M^{e \cdot d} = M \pmod{n}$$

Also we require any message 1st decoded and then encoded be the original message, we have:

$$\left(M^{d}\right)^{e} = M^{e \cdot d} = M \pmod{n}$$

These two requirements ensure the mapping between plain text space and cipher text space will be one to one on onto. If a function has these two properties, then the function is bijective. The has two immediate consequences. Firstly bijective functions have inverses and secondly a composition of bijective functions is also bijective. Hence one may cascade multiple encryptions without losing information. Being able to use a composition of public key encryption functions, enables one to digitally sign documents.

RSA's bijectivity amounts to its being a type of monoalphabetic substitution cipher. Although in practice the "alphabet" will have n "letters" where n has thousands of digits.

Now for review, let's look at our earlier identity due to Euler.

$$M^{k \cdot \phi(n)+1} \equiv M \pmod{n})$$

And our encryption/decryption requirement

$$M^{e\cdot d} = M \pmod{n}$$

By simply equating the exponents, we find

$$e \cdot d \equiv k \cdot \phi(n) + 1$$

This statement gives the needed relationship between the encode exponent, decode exponent and the prime factors of *n*. Notice how an attack on RSA requires one to be able find *d* given *e* and *n*. Putting in the specific value for the totient, we find for the exponent relation:

$$e \cdot d \equiv k \cdot (p-1)(q-1) + 1$$

This form explicitly shows how one needs the prime factors of *n* to find *d*.

How Secure is RSA?

When Martin Gardner 1st described the RSA algorithm in his column in *Scientific American* magazine in 1977, he forwarded a challenge from RSA's authors to break a moderate example of the RSA method. This was finally broken in 1993-1994 using 600 machines running for 6 months all just to factor a 129 digit number so as to be able to decipher the code. The secret message was "The Magic Words are Squeamish Ossifrage."

Rivest using what was known, in 1977, about factoring concluded that breaking his example would require 40 quadrillion years. Needless to say advances in factoring have reduced this time. But factoring numbers with thousands of digits are currently out of reach of known methods even though finding large primes and forming products are well within the reach of home desktop machines. But who knows what advances in computing will bring forward. Some have suggested that quantum computing with its extreme parallelism may be a way to crack the code. But such computing methods are in their infancy and we'll have to see what they grow into.

Simple Example of RSA

To keep the example tractable, we will use small numbers. In a secure situation, one uses very large numbers.

We first generate the keys, so we pick two random primes (normally very large and unequal in length). We will choose 3 and 7 for our example. So this means $n = 3 \times 7 = 21$. Then we compute the totient, $\varphi = \varphi(21) = \varphi(3) \cdot \varphi(7) = 2 \times 6 = 12$. Now let's pick e to be coprime with respect to the totient and with respect to *n*, so we may use 5, 11 etc. We find d=5 when e=5 and d=23 when e=11. Certainly the symmetrical case would not be

used in a public key situation, so we will work with the e=11, d=23 case. Now our plain text value may range from 0 to 20. So we find the following mapping:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
0	1	11	12	16	17	6	7	8	18	19	2	3	13	14	15	4	5	9	10	20

The 1st row contains the possible plaintext values and the second row shows their corresponding cyphertext values. For example "3" encodes as "12." To calculate this simply find $3^{11} (\mathrm{mod}\, 21) = 12$. Looking at the table, we can certainly see the mapping is bijective. By using larger primes, the patterns of regularity will go away. And as mentioned earlier, primes with lengths of 1000s of digits are normally employed.

Pseudo Random Number Generation

Certainly one may see from the mapping that a counter starting at 0 and going up to N-1, that the encrypted version of each value from the counter can be used as a pseudo random number. The set of numbers will have a period of N and a uniform density.

Instead of using a counter and piping its value into the modular exponentiation black box each time, we may feed the last encrypted value back into the box to create a new value. If you examine the previous mapping you will see it will lockup into very short cycles when feeding its output into its input.

But with the right coefficients, the cycles can be very long indeed. So we let e=2 and p and q be congruent to 3 (mod 4), and choose gcd(p-1,q-1) to be small, then we will get long period sequences[4]. This method, called Blum Blum Shub, after its inventors is useful for making cryptographic sequences. The computational overhead would preclude its use in Monte Carlo type simulations. The Mersenne Twister appears to reign in that arena. But linear feedback based approaches are easy to break (hence become predictable for all eternity) once a handful of numbers are observed and therefore become unsuitable for cryptographic applications. Quite often in secure applications one just uses the least significant bit of each output as their cryptographic sequence.

A common form (chosen to have simple exponentiation) of BBS is simply done as follows:

$$x_{n+1} = (x_n)^2 (\mathrm{mod}) N$$

Where N=p*q and p and q are chosen according to the aforementioned rules.

Appendix A – Proof of Fermat's Little Theorem

Fermat says that when *a* and *p* are coprime and p is prime, then $a^{p-1} \equiv 1 \pmod{p}$.

To see this let's write out a list of *p-1* consecutive multiples of *a*, thus we have:

{a, 2a, 3a, 4a, …, (p-1)a}

If $m \cdot a \equiv n \cdot a \pmod{p}$, then $m \equiv n \pmod{p}$. So then our list is equivalent to 1,2,3,4,…,(p-1) (mod p) with a scrambled order. The coprime requirement ensures each value appears only once

Also the products of the terms in both lists are also congruent modulo p. So this means

$a^{p-1}(p-1)! \equiv (p-1)! \pmod{p}$ which after factoring out the factorial

$a^{p-1} \equiv 1 \pmod{p}$ case 1

Also by multiplying both sides by a, we arrive at:

$a^{p} \equiv a \pmod{p}$ case 2

And in case 2, *a* and *p* no longer need to be coprime.

For example: let a=3 and p=3, 3^3 = 27 (mod 3) = 0 = 3 (mod 3) = 0.

Another case 2 example with a and p coprime: a=2 and p=11, thus 2^11 = 2048 (mod 11) = 11*186+2 (mod 11) = 2 (mod 11).

And using the same values in case 1 we find: 2^10 = 1024 (mod 11) = 93*11+1 (mod 11) = 1 (mod 11).

Appendix B – Dirichlet[12] Convolution

The theory of totients is rooted in the theory of arithmetic functions, where their properties become defined under convolution. I will give here some definitions and properties about such functions without proof. One may study [5] to get the actual details.

Arithmetic functions are defined over the domain of positive integers and have a range that extends over a subset of the complex numbers. If given two arithmetic functions we can define a new arithmetic function in terms of the original two. For example, let our two functions be $f(n)$ and $g(n)$, then we define their Dirichlet convolution product or simply "product" as given by:

$$f \cdot g(n) = \sum_{d|n} f(d) \cdot g\left(\frac{n}{d}\right)$$

where the summation is over all positive "d" that divides "n."

We can define an "identity" function, $\varepsilon(n) = 1$ when $n = 1$ and equals zero otherwise. A subset of the arithmetic functions called "units" are arithmetic functions that have inverses such that their convolution products result in the identity function, e.g.,

$$\varepsilon(n) = f \cdot f^{-1}(n)$$

A necessary and sufficient condition for an arithmetic function to have an inverse is simply that its value at n=1 be nonzero.

A subset of units contains the multiplicative functions. These have the property that $f(1) = 1$ and $f(a \cdot b) = f(a) \cdot f(b)$. If this is true for all a,b in the positive integers, then f() is completely multiplicative. If it is true only for a and b coprime, then f() is simply multiplicative.

The set of iota functions of order, s (s may be complex), is defined to be $\iota_s(n) \equiv n^s$. Now we can define a function $f(n)$ to be a totient if and only if

$$f(n) = g \cdot h^{-1}(n)$$

where $g(n)$ is completely multiplicative and $h^{-1}(n)$ is the inverse of a completely multiplicative function. Note: the inverse of a completely multiplicative function does not have to be completely multiplicative.

---

[12] Johann Peter Gustav Lejeune Dirichlet [1805 - 1859] , A German mathematician who gave us the formal definition of a function. He also made many valuable contributions to number theory.

For the case of the Euler totient we have $\varphi(n) = \iota_1 \cdot \iota_0^{-1}(n)$. The inverse of the iota zero function is also known as the Mobius function.

The following table shows the aforementioned arithmetic functions with their valuations for n=1 up to 20. Columns 2 to 4 are completely multiplicative functions and columns 5 and 6 are just multiplicative. Column 5 which contains the Mobius function can also be described as being equal to one when n=1, equal to 0 whenever n contains as a factor a square of a prime and equal to (-1)^k when n contains k distinct prime factors.

Table of some arithmetic functions for n=1 up to 20					
n	$\varepsilon(n)$	$\iota_0(n)$	$\iota_1(n)$	$\iota_0^{-1}(n)$	$\varphi(n)$
1	1	1	1	1	1
2	0	1	2	-1	1
3	0	1	3	-1	2
4	0	1	4	0	2
5	0	1	5	-1	4
6	0	1	6	1	2
7	0	1	7	-1	6
8	0	1	8	0	4
9	0	1	9	0	6
10	0	1	10	1	4
11	0	1	11	-1	10
12	0	1	12	0	4
13	0	1	13	-1	12
14	0	1	14	1	6
15	0	1	15	1	8
16	0	1	16	0	8
17	0	1	17	-1	16
18	0	1	18	0	6
19	0	1	19	-1	18
20	0	1	20	0	8

References

[1] R. Rivest, A. Shamir, L. Adleman, “A Method for Obtaining Digital Signatures and Public Key Cryptosystems” *MIT/LCS/TM-82*, Apr 1977
[2] J. Sylvester, “On Certain Ternary Cubic Form Equations”, *Amer. J. Math*. 2 (1879) pp 280-285, 357-393
[3] W. Diffie, M. Hellman,”New Directions in Cryptography”, *IEEE Trans. On Information Theory*,pp 644-654, Nov 1976
[4] L. Blum, M. Blum, M. Shub, “A Simple Unpredictable Pseudo-Random Number Generator”, *SIAM Journal on Computing*,vol 15, pp 364-383, May 1986.
[5] A. A. Gioia, The Theory of Numbers, Dover Publications, New York, 2001