Algologic Technical Report #1/2012

Multiplicative inverse in mod(m)[1]

S. Parthasarathy
drpartha@gmail.com

Abstract

This is a tutorial on an important aspect of modular arithmetic. Modular arithmetic finds several uses in cryptology. Although a very simple concept, it has very profound mathematical implications. This report examines the concept of multiplicative inverse in modular arithmetic, using various examples. A companion paper [5] studies the group theoretic properties of modular arithmetic.

Ref.: euclid1.tex
Version : 20120229a

1 Basic definitions

In the text that follows, we use the symbols "*" and "." (ignore the quote signs) to denote traditional arithmetic multiplication.

$\mathbb{Z}$ Denotes the set of integers {..., -2, -1, 0, 1,2,3,4,....}

$\mathbb{Z}^+$ Denotes the set of positive integers {1,2,3,4,....}

$\mathbb{Z}_m$ Denotes the set of residues (remainders) {0,1,2,3,4,5,.....m-1} when $z \in \mathbb{Z}$ is divided by m.

[1] This report is accessible on the web at http://www.freewebs.com/profpartha/publications/downloadables.htm#maths. The LaTeX source of this document, as well as this rendered file (pdf version), may be obtained by sending a request to drpartha@gmail.com. Please quote the Reference code and the Version code given above.

2 The multiplicative inverse problem

Given $P,\ find\ Q\ such\ that\ P*Q=1$ is a trivially simple problem in elementary arithmetic. e.g. Given $P=5, P*Q=1 \Longrightarrow Q=1/5$

Q is said to be the *multiplicative inverse of P*, and vice-versa.

This same problem may be restated in modular arithmetic as :

Problem statement :: Given $P,m\ ,\ find\ Q\ such\ that\ P*Q \equiv 1 * mod(m)$

Q is said to be the *modular multiplicative inverse of P*, and vice-versa. Computing Q in mod(m) arithmetic is not as stratightforward as in elementary arithmetic. Two approaches are possible ::

- Brute-force, or trial-and-error method
- Use of Extended Euclidean algorithm

We shall use the trial-and-error method on a small example. This method becomes cumbersome for larger values of P, Q or m. In which case, the Extended Euclidean algorithm comes in handy.

Given P=3 and m=11, find the multiplicative inverse of P. In other words, find Q such that $Q*3 \equiv 1 mod(11)$ By trial-and-error, we can find that the smallest integer which solves this congruence is 4 because $3*4=12 \equiv 1 mod(11)$

Now, let us find the inverse mod(3220) of 79. In other words, find Q such that Q*79=1 mod(3220). The trial-and-error method we used earlier wo'nt be easy, unless we are exceptionally lucky.

3 Conditions for existence of modular multiplicative inverse

Problem statement : Given $P,m\ ,\ find\ Q\ such\ that\ P*Q \equiv 1 * mod(m)$

A sufficient condition for the existence of Q is that P and m be mutually coprimes i.e. gcd (P,m) = 1. In other words, an integer P does not have an multiplicative inverse under modulo m, if P and m have a common factor > 1 i.e. $gcd(P,m) > 1$.

The proof of this statement is left as an exercise for the reader (YOU !)

Any positive integer that is less than n and relatively prime to n has a multiplicative inverse modulo n. This is a consequence of the Euclidean algorithm. Any positive integer that is less than n and not relatively prime to n does not have a multiplicative inverse modulo n. gcd(15, 26) = 1; 15 and 26 are relatively prime. Therefore, 15 has a multiplicative inverse modulo 26.

4 Euclid's algorithm

The Euclidean algorithm is one of the oldest algorithms still in common use [1]. It appears in Euclid's Elements (c. 300 BC), specifically in Book 7 (Propositions 12) and Book 10 (Propositions 23). In Book 7, the algorithm is formulated for integers, whereas in Book 10, it is formulated for lengths of line segments. (In modern usage, one would say it was formulated there for real numbers.)

Euclid's theorem may be stated as:

$$If\ a\ and\ b\ are\ integers\ and\ a > b,\ then\ gcd(a,b) = gcd(a(modb),b) \quad (1)$$

The proof of validity of this theorem can be found in [1]

Euclid's algorithm stated above (circa 300 BCE approx) is used for computing the gcd of two integers, using a simple iterative procedure.
Example: To compute the gcd of 259 and 70, we use the Euclidean algorithm :

$$\begin{aligned} gcd(259,70) &= gcd(259 mod 70, 70) \\ &= gcd(70,49) && (2) \\ &= gcd(70 mod 49, 49) \\ &= gcd(49,21) && (3) \\ &= gcd(49 mod 21, 21) \\ &= gcd(21,7) && (4) \\ &= gcd(21 mod 7, 7) \\ &= gcd(7,0) && (5) \\ &= 7 && (6) \end{aligned}$$

259 = 3(70) + 49	21 = 70 - 1(49)
70 = 1(49) + 21	7 = 49 - 2(21)
49 = 2(21) + 7	= 49 - 2(70 49)
21 = 3(7) + 0	= 3(49) - 2(70)
	= 3(259 -3(70)) 2(70)
	= 3(259) - 9(70) - 2(70)
	= 3(259) - 11(70)

So gcd(259, 70) = 7 = 3(259) - 11(70).

5 Extended Euclid's Algorithm

If a and b are positive integers, then there are always integers m and n so that the GCD of a and b equals ma+nb.

$$ma + nb = gcd(a, b) \tag{7}$$

The above expression is also known as *Bezout's identity* (named after the French mathematician Étienne Bézout). The above equation does not have a unique solution. For instance, the greatest common divisor of 12 and 42 is 6. Bézout's identity states that there must exist an integer solution for x and y in the following equation:

$$12x + 42y = 6. \tag{8}$$

One of its solutions is x = -3 and y = 1: indeed, we have (-3)*12 + 1*42 = 6. Another solution is x = 4 and y = -1.

This class of equations which involves only integer solution is called a Diophantine equation [8]. When the equation is a linear equation, we have a Linear Diophantine equation.

The Extended Euclidean algorithm not only computes gcd(a,b), but also returns the numbers m and n such that $gcd(a, b) = a * m + b * n$ [2][7]. In other words, the gcd of any two integers a and b, can be written as a linear combination of a and b. We use this property to compute the multiplicative inverse of a number. If gcd(a,b)=1 this solves the problem of computing modular inverses.

Observation : The extended Euclidean algorithm $gcd(a, b) = a * m + b * n$ is particularly useful when a and b are coprime, since **m is the multiplicative inverse of a modulo b, and n is the multiplicative inverse of b modulo a.** (Proof left to the reader as an exercise)

Given an element $a \in \mathbb{Z}_m$, its inverse can be computed by using the Euclidean algorithm to find gcd(a, m), since that algorithm also provides a solution to the equation $ax + my = gcd(a, m) = 1$, which is equivalent to $ax \equiv 1(mod\ m)$. In other words, assume that we want to compute $n^{(-1)}\ (mod\ m)$ and further assume that gcd(n,m)=1. Run the Extended Euclidean algorithm to get a and b such that a*n+b*m=1. Rearranging this result, we see that $a * n = 1 - b * m$, or $a * n = 1(mod\ m)$. This solves the problem of finding the modular inverse of n, as this shows that $n^{(-1)} = a(mod\ m)$.

The Extended Euclidean algorithm is nothing more than the usual Euclidean algorithm, with side computations to keep careful track of what combination of the original numbers n and m have been added at each step.

Example : Find the multiplcative inverse of 20 mod 97. The extended Euclidean algorithm allows us to write 1 as a linear combination of 97 and 20. Here we go:

17=1*97-4*20
20-1*17=3 so 3=1*20-1*17=1*20-(1*97-4*20)=-1*97+5*20
17=5*3+2 so 2=17-5*3=(1*97-4*20)-5(-1*97+5*20)=6*97-29*20
1=3-2=(-1*97+5*20)-(6*97-29*20)=-7*97+34*20

The final equation tells us that 1=-7*97+34*20, which means that the product of 34 and 20 is equal to 1 plus a multiple of 97. But in mod 97, we ignore multiples of 97. Therefore 34 is the multiplicative inverse of 20 mod 97.

Check back :: 34*20=680=679+1=7*97+1=1mod97. QED

Example: To find the multiplicative inverse of 17 in Z_{64}, we use the Extended Euclidean algorithm:

$$64 = 3.17 + 13 \rightarrow r = 13 \tag{9}$$
$$17 = 1.13 + 4 \rightarrow r = 4 \tag{10}$$
$$13 = 3.4 + 1 \rightarrow r = 1 \tag{11}$$
$$4 = 4.1 + 0 \rightarrow r = 0 \tag{12}$$

Notice that 17 and 64 are relatively prime. Now we can compute the multiplica-

tive inverse of 17 (or 64) by working backward:

$$\begin{aligned} 1 &= 13 - 3.4 && (13)\\ &= 13 - 3.(17 - 1.13) && (14)\\ &= 4.13 - 3.17 && (15)\\ &= 4.(64 - 3.17) - 3.17 && (16)\\ &= 4.64 - 15.17 && (17) \end{aligned}$$

Hence (15) . $17 \equiv 1(mod 64)$, but $15 \equiv 49(mod 64)$, so the inverse of 17 in Z_{64}, is 49. We will denote this by writing $17^{-1} = 49(mod 64)$, or $17^{-1} mod 64 = 49$.

Check back :: 17*49=833=832+1=13*64+1=1mod 64 QED

Problem statement : Find the multiplicative inverse of 79 mod(3220). i.e. find Q such that $Q * 79 \equiv 1\ mod(3220)$.

First let us compute gcd(79,3220) using Euclidean algorithm:

$$\begin{aligned} gcd(79, 3220) &= gcd(3220 mod 79, 79) && (18)\\ &= gcd(79, 60) && (19)\\ &= gcd(79 mod 60, 60)\\ &= gcd(60, 19) && (20)\\ &= gcd(60 mod 19, 19)\\ &= gcd(19, 3) && (21)\\ &= gcd(19 mod 3, 3)\\ &= gcd(3, 1) && (22)\\ &= gcd(3 mod 1, 1)\\ &= gcd(1, 0) && (23)\\ &= 1 && (24) \end{aligned}$$

Using the extended euclidean algorithm, equations 23 and 18 can be rewritten as :

$$79 * x + 3220 * y = gcd(79, 3220) = 1 \quad (25)$$

60	79	$(1, -40)$	$(0, 1)$
60	19	$(1, -40)$	$(-1, 41)$
3	19	$(4, -163)$	$(-1, 41)$
3	1	$(4, -163)$	$(-25, 1019)$
0	1	$(79, -3220)$	$(-25, 1019)$

In fact, we can also use the online modulo inverse calculator from Princeton University [6], which gives modular multiplicative inverse of 79 mod(3220) as 1019.

Check back :: 1019*79= 80501= 1 + 25*3220 = 1mod(3220) QED

6 An application of multiplicative modulo inverse

[4] gives an overview of the use of number theory in cryptography.

In cryptography, Caeser's cipher consists of creating a secret message (encrypted text, or cipher text), by shifting each letter of the English alphabet (in the plain text) by k positions in the English alphabet. Thus, for k=3, A becomes E, B becomes F, C becomes G, and so on. We can imagine that the alphabet is written on a clock face with 26 places. So, W becomes A, X becomes B, Y becomes C, Z becomes D.

This simple cryptographic protocol may be extended to include all letters of the English alphabet (upper case, and lower case), and also numerals and special characters i.e. entire ASCII charcter set (of 256 characters), as follows.

Encoding

The encrypted character y is obtained from the plain text character x using the formula $y = x + k \ (mod\ 256)$ simply shift the character set $k\ mod\ 256$ places away.

A more sophisticated approach is to use a more general linear equation such as the following to encode data $y = ax + \ b \ (mod\ m)$

Decoding

As long as the coefficient, a, and the modulus, m, are relatively prime, the equation may be solved for x yielding the decryption equation, $x = a^{-1} * (y - b)\ (mod m)$ where a^{-1} is the multiplicative inverse of a in mod m.

7 Acknowledgements

The author thanks Dr. K. Srinathan, of IIIT-Hyderabad, for explaining him the Extended Euclidean Algorithm.

Dr. Kumar Eswaran reviewed the draft report and helped me remove several flaws.

8 Concluding Remarks

This report was typeset in LaTeX by the author (using Kile, on a Suse Linux system). The LaTeX source can be obtained from the author, by sending a request to drpartha@gmail.com.

Writing a mathematically rich document, and typesetting it using LaTeX , is not easy. The author welcomes comments, suggestions and remarks (on this report. Please report any flaws, inconsistencies, and ambiguities you may find, to drpartha@gmail.com. The author thanks you in advance, for this gesture. Every such communication will also be personally acknowledged by the author, and taken into account in the next version of this report.

The author will be happy to answer any questions you may have on this subject.

References

[1] `Wikipedia`
http://en.wikipedia.org/wiki/Euclidean_algorithm#Proof_of_validity

[2] *http://userpages.umbc.edu/ rcampbel/NumbThy/Class/ BasicNumbThy.html#Modular-GCD*

[3] `Wikipedia`
http://en.wikipedia.org/wiki/Extended_Euclidean_algorithm

[4] Karl Petersen, Notes on number theory and cryptography, *math.unc.edu/Faculty/petersen/Coding/cr2.pdf*

[5] `S. Parthasarathy`, *Group theoretic view of modular arithmetic*, Algologic Technical Report #4/2012, Feb. 2012.

[6] `Darren Sri-Jayantha`, http://www.cs.princeton.edu/ dsri/modular-inversion.html

[7] `DI-Management` http://www.di-mgt.com.au/euclidean.html

[8] `Wolfram Mathworld` http://mathworld.wolfram.com/DiophantineEquation.html

* * *$_{\text{end}}$