MysteryTwister C3

THE CRYPTO CHALLENGE CONTEST

Modulo Arithmetic with n – Appendix 1

Authors: Tanja Lange, Andreas Gruener, Bernhard Esslinger

August 2011

Introduction Modulo Arithmetic

Tom's idea, within the presented task, was to use the "Affine Cipher" with $y = a \cdot x + b \bmod 26$ as encryption equation. However, what is the meaning of the mathematical operator "mod"?
The functional principle of "modulo" will be explained within the subsequent slides.

Assignment of Letters to Numbers

According to the task, the Latin alphabet is used to represent messages. Classical shift ciphers, for example Caesar Cipher, can be made clearer while using a cipher disk. Cipher disks contain the alphabet, which is arranged as circle. Thus after the letter Z, the letter A reoccurs.

Coat of arms of the NSA

[1]

Within the presented task a more complex cipher scheme is applicable. Therefore, it is necessary to operate with numbers instead of letters. We map letter A to number 0, letter B to number 1, letter C to number 2, and so on. The numbers from 0 to 25 are structured analog to the arrangement of letters on the cipher disk.

Concerning the Caesar Cipher, it suffices to shift the alphabet for determining a ciphertext. Furthermore the shift of the alphabet can be described as addition from a number in $\{0,1,2,3,\ldots,23,24,25\}$ to the plaintext letter.

In contrast, to calculate a ciphertext letter according to the "Affine Cipher", the plaintext letter gets multiplied with value a and value b will be added. If the result is bigger than 25, the value 26 will be subtracted until the result lies within the range {0,1,2,3,...,23,24,25}.

Other alphabets require more or less than 26 letters. Thus the alphabet pays attention to case-sensitive letters and special characters. In this case the alphabet size n will be subtracted and not 26 if the result is bigger or equal n. Henceforward we see the alphabet as numbers $\geqslant 0$ and $< n$. The assignment of letters to numbers and the needed arithmetic as well as the mathematical foundation will be described in more detail in the subsequent part.

Remainders, Equivalence and the „Modulo“ Operator

Integers can be used to compute various mathematical operations, for example addition and multiplication. Let s be a random number which is the result of several mathematical operations. We choose the remainder of the division s/n as unambiguous element of the alphabet for s. The equation $s = r \cdot n + t$ represents s for integers r and t with $0 \leqslant t < n$. We choose t as result and write:

$$s \equiv t \bmod n.$$

Mathematicians declare this kind of calculus with remainders regarding a fixed integer n as “modulo arithmetic”. One might say that s is equivalent to t modulo n if the resulting remainders of division s/n and t/n are equal.

Calculations can be illustrated on a disk where the numbers from 0 to $n-1$ are consecutively printed on the edge (so that after 0 comes $n-1$). Let s and t be random numbers between 0 and $n-1$: $s \cdot t$ means that t is added s-times to itself. In other words, the value on the disk is rotated s-times with t-steps. We can respectively express the exponentiation s^t as multiplying s t-times with itself. It is helpful to keep intermediate results small. The following algorithms, which can be proofed, are very useful for calculations.

$$
\begin{aligned}
s + t &\equiv (s \bmod n) + (t \bmod n) \bmod n \\
s - t &\equiv (s \bmod n) - (t \bmod n) \bmod n \\
s \cdot t &\equiv (s \bmod n) \cdot (t \bmod n) \bmod n
\end{aligned}
$$

So, you will get the same result if you already use numbers of the interval (0,n-1) in between.

The Division

According to "normal" integers, division is the inverse operation of multiplication. To calculate $s = 1/t \bmod n$, we search a number $0 \leqslant s < n$ which results in 1 by multiplication with t. A complete search for small numbers n is possible without problems. For instance, we can create a complete multiplication table. The following figure presents such a table for $n = 5$.

*	0	1	2	3	4
0	0	0	0	0	0
1	0	1	2	3	4
2	0	2	4	1	3
3	0	3	1	4	2
4	0	4	3	2	1

The inverse element of $s = 3$ modulo $n = 5$ can be looked up as $t = 2$; because $s \cdot t = 2 \cdot 3 = 6 \equiv 1 \bmod 5$.

If you create such a table for $n = 6$, you get:

*	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	1	2	3	4	5
2	0	2	4	0	2	4
3	0	3	0	3	0	3
4	0	4	2	0	4	2
5	0	5	4	3	2	1

If you seach the inverse element of $s = 3$, you notice that there is no number t with $s \cdot t = 1$ modulo 6. This also occurs for $s = 2$ and $s = 4$.

If you investigate this phenomenon for other values of n, you can determine, that this is happening if n has pure factors. If this is the case, inverse elements of numbers which have a common factor with n do not exist. Thus, you can postulate a necessary and sufficient condition for s having an inverse modulo n. The number s has an inverse element modulo n if and only if the greatest common divisor of s and n is 1 (gcd(s,n)= 1). Therefore, all elements except 0 have an inverse element for a prime n.

Mathematicians would say that the set natural numbers, which are in the range from 0 to a certain prime n, plus specific mathematical operations and particular rules build a "finite field". Within this range of numbers, each number has an appropriate multiplicative inverse element.

How can a multiplicative inverse be determined? It is not possible to create a multiplication table for really big values of n, for example $n \approx 10^{300}$. For this case an algorithm called "Extended Euclidean Algorithm" is auxiliary. This algorithm computes gcd(s,n) for input s and n as well as the inverse of s modulo n if both numbers are relatively prime.

The Extended Euclidean Algorithm

The extended Euclidean algorithm starts with s and n as input. The result is the greatest common divisor as well as two numbers u and v. The number u represents the multiplicative inverse of s modulo n.

$$\gcd(s, n) = u \cdot s + v \cdot n$$

The first step of computation encompasses the search of an integer multiplier w_1 and an integer remainder z_1 which permit the illustration of n with s in the following way: $n = w_1 \cdot s + z_1$. During the next step, s is shifted to the position of n and z_1 is shifted to the position of s. An integer multiplier w_2 as well as an integer remainder z_2 for the equation $s = w_2 \cdot z_1 + z_2$ is searched again. The steps of calculation will be repeated until the remainder is 0.

$$
\begin{gathered}
n = w_1 \cdot s + z_1 \\
s = w_2 \cdot z_1 + z_2 \\
z_1 = w_3 \cdot z_2 + z_3 \\
\cdots \\
z_{m-2} = w_m \cdot z_{m-1} + z_m \text{ mit } z_m = 0
\end{gathered}
$$

If the remainder is 0, the sums will be backwards inserted in each other and combined.

$$
\begin{aligned}
z_{m-1} &= z_{m-3} - w_{m-1} \cdot z_{m-2} \\
&= z_{m-3} - w_{m-1} \cdot (z_{m-4} - w_{m-2} \cdot z_{m-3}) \\
&= \cdots \\
&= u \cdot s + v \cdot n
\end{aligned}
$$

The following example should clarify the functional principle of the extended Euclidean algorithm.

The multiplicative inverse of 26 modulo 57 should be calculated. Therefore, $s = 26$ and $n = 57$.

$$57 = 2 \cdot 26 + 5$$
$$26 = 5 \cdot 5 + 1$$
$$5 = 5 \cdot 1 + 0$$

$$1 = 26 - 5 \cdot 5$$
$$1 = 26 - 5 \cdot (57 - 2 \cdot 26) = 26 - 5 \cdot 57 + 10 \cdot 26 = 11 \cdot 26 - 5 \cdot 57$$

The result of the computation can be read off. The multiplicative inverse of 26 modulo 57 is 11.

$$\gcd(26,57) = 11 \cdot 26 - 5 \cdot 57$$

An elaborate description of the extended Euclidean algorithm as well as further examples can be found on Wikipedia in the appropriate article [2].

References

[1] National Security Agency: Coat of arms of the National Security Agency. URL: http://www.nsa.gov

[2] Extended Euclidean Algorithm. Wikipedia. URL: http://en.wikipedia.org/wiki/Extended_Euclidean_algorithm