# CMSE-512 Database and File Security

# Problem Session 04.04.2025

[Security of Information Systems](https://staff.emu.edu.tr/alexanderchefranov/Documents/CMSE512/Spring%202022/Security%20of%20Information%20Systems%2028022022.docx), [Protection of passwords](https://staff.emu.edu.tr/alexanderchefranov/Documents/CMSE512/Spring%202022/Protection%20of%20Passwords1.doc), [RSA algorithm](https://staff.emu.edu.tr/alexanderchefranov/Documents/CMSE512/Spring2023/RSA%20algorithm.docx), [DES](https://staff.emu.edu.tr/alexanderchefranov/Documents/CMSE512/Spring%202022/DES.docx),  [SHA512](https://staff.emu.edu.tr/alexanderchefranov/Documents/CMSE512/Spring%202022/SHA512%2028032022.docx)

Cryptography, both symmetric and asymmetric, widely uses number theory, in particular, relative primality, modular arithmetic, and multiplicative inverses. That’s why we begin with them.

1. Primality, Greatest Common Divisor (GCD), Euclidean Algorithm

Prime number is one having no other factors except one and itself, e.g. N=7 is a prime number

Relatively prime are such two numbers that have no common factors except one

Greatest Common Divisor (GCD) is the maximal common factor for two numbers

For example, gcd(12,15)=3

GCD can be obtained by factoring the numbers and comparing them

For example, 12=2x2x3, 15=3x5, hence, gcd(12,15)=3

Euclidean algorithm provides straightforward method of finding gcd without necessity of finding factors

EUCLID(a,b)

1. A:=a; B:=b
2. if B=0 return A=gcd(a,b)
3. R=A mod B
4. A:=B
5. B:=R
6. goto 2

The algorithm has the following progression:

A1=B1xQ1+R1

A2=B2xQ2+R2

A3=B3xQ3+R3

*To find gcd(1970,1066)*

*1970=1x1066+904 gcd(1066,904)*

*1066=1x904+162 gcd(904,162)*

*904=5x162+94 gcd(162,94)*

*162=1x94+68 gcd(94,68)*

*94=1x68+26 gcd(68,26)*

*68=2x26+16 gcd(26,16)*

*26=1x16+10 gcd(16,10)*

*16=1x10+6 gcd(10,6)*

*10=1x6+4 gcd(6,4)*

*6=1x4+2 gcd(4,2)*

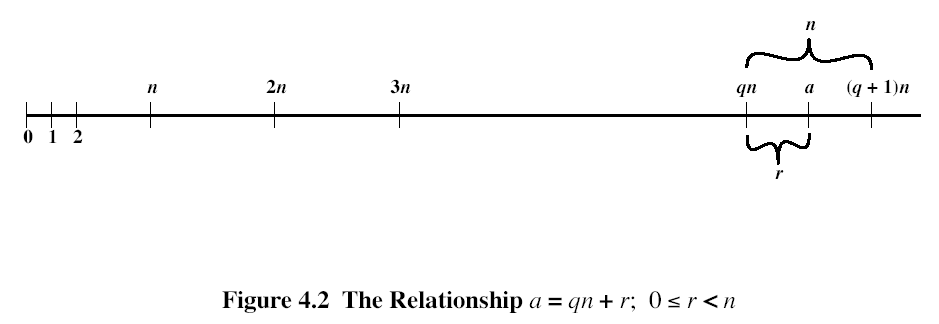
*4=2x2+0 gcd(2,0)*

*Therefore, gcd(1970,1066)=2*

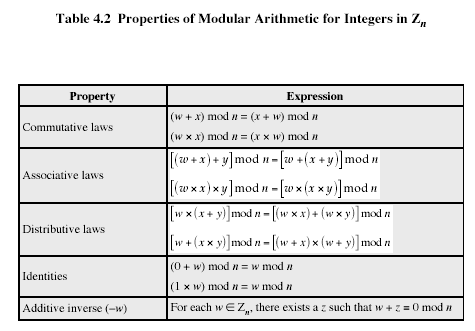
Given any positive integer n and any integer a, if we divide a by n, we get an integer quotient q and an integer remainder r that obey the following relationship:

a=qn+r 

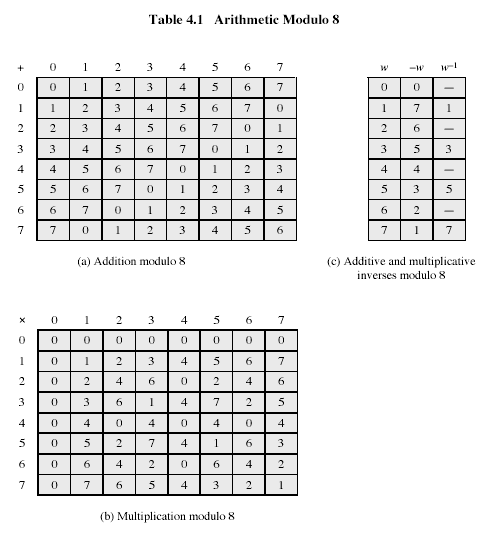
where  is the largest integer less than or equal to x.



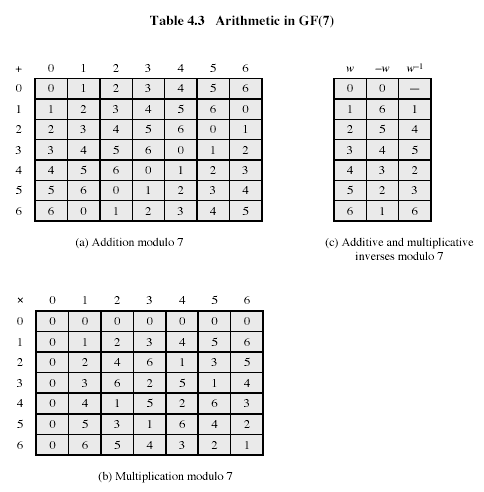
The remainder r is often referred to as a residue. Let Zn ={0,1,..,n-1}.



In general, an integer has a multiplicative inverse in Zn, if that integer is relatively prime to n. Table 4.1c shows that the integers 1, 3, 5, and 7 have a multiplicative inverse, but 2, 4, and 6 do not.



1. Multiplicative inverse, Extended Euclid



If gcd(m,b)=1, then b has a multiplicative inverse modulo m. That is, for positive integer b<m, there exists a b-1<m such that b b-1=1 mod m. Euclid’s algorithm can be extended so that, in addition to finding gcd(m,b), if the gcd is 1, the algorithm returns the multiplicative inverse of b.

EXTENDED EUCLID(m,b)

1. (A1,A2,A3):=(1,0,m); (B1,B2,B3):=(0,1,b);
2. if B3=0 return A3=gcd(m,b); no inverse
3. if B3=1 return B3 = gcd(m,b); B2= b-1 mod m
4. Q=
5. (T1,T2,T3):=(A1-QB1, A2-QB2, A3-QB3)
6. (A1,A2,A3):= (B1,B2,B3)
7. (B1,B2,B3):= (T1,T2,T3)
8. goto 2

Throughout the computation, the following relationships hold:

mT1+bT2=T3 mA1+bA2=A3 mB1+bB2=B3

To see that algorithm correctly returns gcd(m,b), note that if we equate A and B in Euclid’s algorithm with A3 and B3 in the extended Euclid’s algorithm, then the treatment of the two variables is identical. Note also that if gcd(m,b)=1, then on the final step we would have B3=0 and A3 =1. Therefore, on the preceding step, B3=1. But if B3=1, then we can say the following:

mB1+bB2=B3

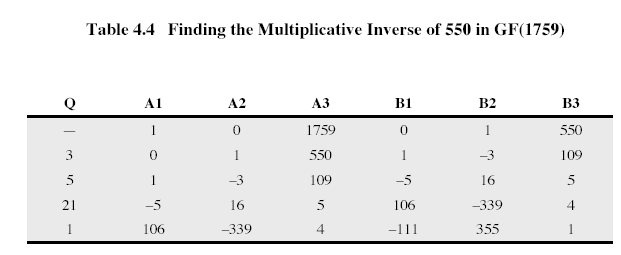
mB1+bB2=1

bB2=1-mB1

bB21 mod m

Hence, B2 is the multiplicative inverse of b.

Table 4.4 is an example of the execution of the algorithm. It shows that gcd(550,1759)=1 and that the multiplicative inverse of 550 is 355; that is, 550x3551 mod 1759.



1. What are the four main security requirements?
2. What is the difference between a threat and vulnerability?
3. What are the four computer system assets?
4. Types of attacks? What are passive attacks? What are active attacks?
5. What are assurance, anonymity and authenticity?
6. How does virtual memory support memory protection? How is provided separation of processes?
7. What is access control? What types of access control do you know? What is user-oriented access control? What is data oriented access control? What is access control matrix? What is access control list? What is capability list (ticket)? How these are related to each other?
8. What are the three types of intruders?
9. What technique is used for intrusion?
10. Why “salt” is kept in clear in Unix password scheme?
11. What are password selection strategies and what are their deficiencies?
12. How intrusion can be detected?
13. What is audit? What information is usually kept in audit file records?
14. What is malicious software? What are the main virus types (Parasitic, Memory-resident, Boot sector, Stealth, Polymorphic, Macro)? What are the life steps of a virus?
15. RSA: Settings of RSA, use of prime numbers
16. RSA: Public and private keys generation, Euler totient function, condition satisfied by the pair of private and public keys
17. RSA: Finding RSA keys using Extended Euclid Algorithm
18. RSA: RSA encryption and decryption; efficient exponentiation using 1) exponent binary decomposition; 2) squaring; 3) modulo reduction
19. DES: Overall schema, inputs, outputs
20. DES: Initial permutation, inverse of initial permutation
21. DES: Round structure; input representation; processing of halves
22. DES: Structure of the round function F
23. DES: Expansion-permutation
24. DES: S-boxes
25. DES: Permutation function P
26. DES: Feistel structure; decryption
27. DES: Round key generation, Permuted choice, Permuted choice 2, Schedule of shifts
28. SHA-512: Structure, padding, initial values, elementary function 80-round structure, addition modulo 2^64, round inputs, words Wi and constants Ki, round structure, functions Ch, Maj, Sum1, Sum0. Generation of Wi and Ki.