* Computer and network security involves many different aspects of providing a safe, reliable environment for computer and network users
* The “big three” characteristics of a secure system are realized in the acronym CIA
  + Confidentiality
  + Integrity
  + Availability
* Confidentiality is the property of keeping data secret between communicating parties
* Integrity
  + Origin integrity (authentication): proof that information originated with a specific individual
  + Data integrity: data is free from tampering, and appears as it originated from the sender
* Availability is the property in which resources are there (available) when you need them
* Cryptography is one of the most important areas of computer science as used in business, especially e-commerce, and in transmission of sensitive data in general
  + It is an enabling feature of confidentiality and even integrity in secure systems
  + It involves processing data, converting it into an unintelligible form that no one can read (ideally) except the intended recipient(s) of that data
* As the name suggests, **symmetric key cryptography** involves two communicating parties with copies of the *same key* using it to convert their messages, and communicate over an insecure medium
* Symmetric key encryption is ancient – it has been around for a couple thousand years
  + Often, symmetric key encryption was done with pen and paper (or their analogues) and is often called **classical encryption**
* The ancient Greeks used the scytale (skih-tuh-lee) to encrypt messages
* The **Caesar cipher,** attributed to Gaius Julius Caesar, is an example of a **substitution cipher** where original characters in the plaintext are substituted with other characters to result in the ciphertext
* Caesar used:
* A key of 3
* Encryption cipher: “left shift 3 characters”
* Decryption algorithm: “right shift 3 characters”
* It sounds (and is) incredibly simple by today’s standards, but it was quite effective in keeping data secret should a messenger be captured by the enemy
* Keep the large circle stationary, and move the  
  inner circle to the left (counter-clockwise)  
  for the substitution ciphertext alphabet
* Another symmetric cryptography technique is the **transposition cipher**, which involves shuffling letters *within a message* to perform the encryption
  + This is different from substitution, since substitution performs a complete, but consistent, replacement of letters
* Once of the simplest forms of transposition is the **rail-fence technique**
* The key is the “number of rails”
* It’s easier to see from an example, so let’s see one
  + **Plaintext:** meet me in the lobby
  + **Key:** 2 (so, 2 rows will be used)
* **Algorithm**: put the message in a table with 2 rows in column major order
  + **Column major ordering** means we fill in one column at a time, as opposed to **row major**, which is one row at a time)
  + Then, the ciphertext is produced by reading the message in row major order:

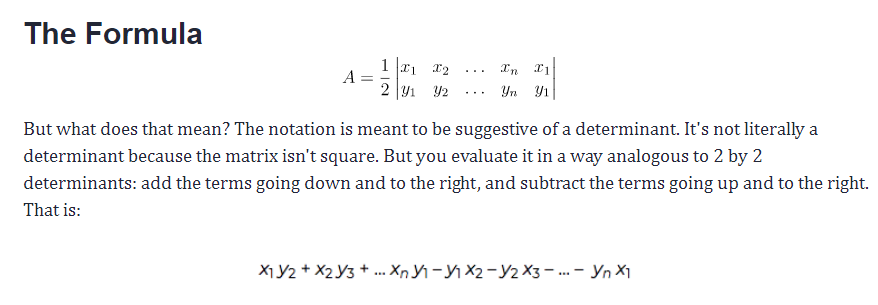
**memiteobetenhlby**

* Decryption is quite easy
* The number of rails (in our case, 2) is the cryptographic key
  + Notice both parties use the same key, but different algorithms: **symmetric** key encryption!
* You apply the algorithm in reverse:
  + Create table with 2 rows
  + Divide number of characters in ciphertext message by the number of rows to obtain the number of columns and take the ceiling (in our case, we have 16 characters and 2 rows, so **ceiling(16/2) = 8**
  + This creates a 2 x 8 table – fill in the characters of ciphertext in row major (left to right), and we read it off in column major since this restored the original encryption table
* The aforementioned techniques are very simple, but they illustrate principles used in encryption today
* The **Advanced Encryption Standard (AES)** is a symmetric key encryption adopted by the US in 2002
* The encryption breaks the message into blocks and uses the key to permute the bits of the message
  + Also, rows shift and columns are swapped
* AES is very fast and remains secure as of this lecture
* Diffie and Hellman proposed the characteristics of a so-called “asymmetric key system” (public key system), largely for solving the purpose of key distribution
* Diffie and Hellman envisioned a system where *two keys* existed: a **public key** and a **private key**
  + The public key could be distributed publicly, or at least without fear of problems if multiple people had it
  + The private keys would not be identical to the corresponding public key, but would be kept secret
* If Alice wants to send Bob a message, M
  + She uses Bob’s public key, PUBto encrypt the message, generating the ciphertext, C
  + That is:  
     C = Encyption(PUB, M)
* Bob would then be the only person who could use his own *private* key PRB to decrypt the ciphertext, C, restoring the plaintext, M
  + M = Decryption(PRB, C)
* This type of system would therefore, provide **confidentiality**
* It could **also** be used to provide **authenticity (origin integrity)** by Alice using her private key to encrypt the original message or ciphertext, and then Bob (or anyone) could use her public key to decrypt it
* They devised the RSA (Rivest-Shamir-Adleman) system at a Passover Seder in 1977, and published a paper on RSA in 1978
* Formally, we say, given a number c 0:
  + a b mod c (a is congruent to b modulo c)
  + If c evenly divides both a and b
  + That is, a is congruent to b, modulo c, or said another way, a and b produce the same remainder when divided by c
  + This is true **iff (a – b) / c is an integer**
* The definition for modular arithmetic that you should memorize for test purposes is as follows:
  + In arithmetic *modulo* *c* we seek to express an integer **x** as follows:

**x = cq + r**

* + Where r must be non-negative
  + q is the quotient
  + r is the remainder
* Theorem 1.4.1 (Elenbogen/Baugh)

**(a \* b) (mod c) = [(a mod c) \* (b mod c)] (mod c)**

* Compute (64 \* 9 \* 10 \* 5) mod 7
* Answer:
* (64 mod 7 \* 9 mod 7 \* 10 mod 7 \* 5 mod 7) mod 7
* Note, 64 mod 7 can be reduced to (8 mod 7 \* 8 mod 7), each of which is 1, so 1 \* 1 = 1
* The 9 mod 7 = 2
* The 10 mod 7 = 3
* The 5 mod 7 = 5
* So, in total:
*  (1 \* 1 \* 2 \* 3 \* 5) mod 7  
  = 30 mod 7
* Fermat’s Little Theorem: Given a prime number, p, and any integer a, then:

**(ap – a) is evenly divisible by p**

* That is,

**ap  a mod p**

* **Corollary:**
* If p is prime, and a is an integer, **and** a is co-prime to p, then:

**(ap-1 – 1) is evenly divisible by p**

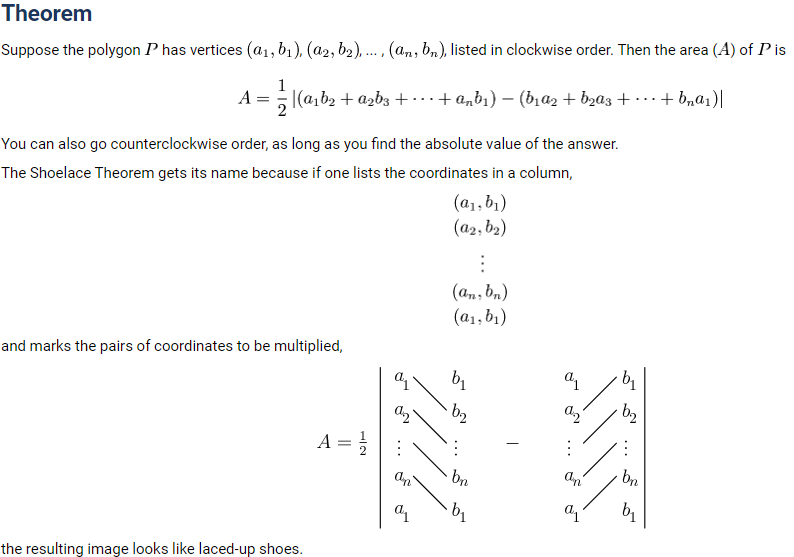
* That is

**ap-1  1 mod p**

* That is, if we divide ap-1 by p, we will have a remainder of 1
* RSA is a block ciper
  + This means the encryption is performed on blocks of plaintext
* Like most public key cryptosystems, there are three distinct steps in RSA:

1. Key generation
2. Encryption
3. Decryption
4. To allow others to securely communicate with her, Alice chooses two primes: p and q and from these, calculates 2 values:
   1. n = pq
   2. Φ(n) = (p-1)(q-1) //phi of n is called the “Euler totient function”
5. Then, Alice chooses an encryption key, e such that gcd(e, Φ(n) )= 1
   1. That is, e and Φ(n) are co-prime
6. Alice calculates the decryption key, d, so that ed 1 (mod Φ(n))
7. Alice makes the public key {n, e} known publicly and keeps her private key {d, p, q} secret

* Bob wants to communicate with Alice

1. Bob looks up Alice’s public key, {n, e}
2. Bob writes his message, m, as m (mod n)
3. Bob computes the ciphertext, c me(mod n)
4. Bob sends c to Alice
5. Alice selects the two primes, p = 17, q = 11
6. She calculates n = pq = (17)(11) = 187
7. She calculates Φ(n) = (p-1)(q-1) = (16)(10) = 160
8. She selects e such that e is relatively prime to Φ(n) = 160
   1. We choose e = 7
9. Determine d such that de 1 (mod 160) , d < 160
   1. d = 23, since 23 \* 7 = 161 = (1 \* 160) + 1
   2. You could have calculated d using Euclid’s algorithm
10. So, PUA = {e, n} = {7, 187}
11. And PRA = {d, n} = {23, 187}
12. We know PUA = {e, n} = {7, 187} and PRA = {23, 187} 🡪 Bob wouldn’t know PRA, just PUA
13. So, if the message M = 88, we calculate ciphertext thusly:
14. C = Me mod n  
     = 887 mod 187  
     = 11
15. Bob sends the C = 11 to Alice
16. Alice receives c from Bob
17. Alice computes the original m c d(mod n)
18. Alice receives the C = 11 from B
19. She computes, using her private key, PRA = {d, n} = {23, 187}:
20. M = Cd (mod n)  
    M = 1123 mod 187  
     = 88
21. So, the original message is retrieved without loss, M = 88

The Euclidean Algorithm for finding GCD(A,B) is as follows:

* If A = 0 then GCD(A,B)=B, since the GCD(0,B)=B, and we can stop.
* If B = 0 then GCD(A,B)=A, since the GCD(A,0)=A, and we can stop.
* Write A in quotient remainder form (A = B⋅Q + R)
* Find GCD(B,R) using the Euclidean Algorithm since GCD(A,B) = GCD(B,R)
* The Euclidean Algorithm states
  + The GCD(m, n) for positive integers m and n, where 0 < m < n is equal to GCD(n mod m, m).
  + When n mod m reaches 0, the algorithm ends and the result is m
* So, what is it good for?
  + Public key systems are useful for solving the original problem they were targeting - key distribution!
  + You initiate communication using a public key system like RSA, and encrypt a symmetric key, which can *then* be used for subsequent communication (in a secret key system like AES)

\*CALCULATE IN CW DIRECTION FOR POSITIVE AREA for surveyors forumla